KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Žydrūnas Naujokas

Miglotai apibrėžtos situacijos įvertinimo modulio sudarymas ir tyrimas

Magistro darbas

Darbo vadovas
prof. R. Jasinevičius

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Žydrūnas Naujokas

Miglotai apibrėžtos situacijos įvertinimo modulio sudarymas ir tyrimas

Magistro darbas

Kalbos konsultantė
Lietuvių k. katedros lekt.
I. Mickienė
2006-05

Vadovas
prof. R. Jasinevičius
2006-05

Recenzentas
Arūnas Žvironas
2006-05

Atliko
IFM-0/1 gr. stud.
Žydrūnas Naujokas
2006-05-15

Kaunas, 2006

TURINYS

RESEARCH AND DEVELOPMENT OF THE MODULE FOR THE EVALUATION OF FUZZILY CHARACTERIZED SITUATION

Summary

The problem of the evaluation of fuzzy situations arises quite often in real life. Images, symbols, signals and so on usually are fuzzily described and can be considered as fuzzy situations. So the parameters, which characterize the fuzzy situation, are not only numerical ones, but verbal too, for example, an object can be described as "heavy, little and moves fast". It is difficult to decide something formally about the object with such characterization. Leasing companies meet this problem in contracts classification process too. Accordingly most leasing companies have own contracts classification (recognition) systems, which mostly are statistically based and the similarities and differences between good and bad contracts are not computed. Therefore the demand arises to develop the intellectual recognition system using fuzzy logic. This system should be simply integrating in any leasing company. It also must compute and measure the similarities and differences between good and bad contracts from their evaluation history. Such intellectual system could be as an adviser for business expert.

The module of fuzzily characterized situation's evaluation is implemented using:

1) fuzzy sets,
2) fuzzy clustering,
3) fuzzy recognition.

The general idea was experimentally investigated using artificially generated data as well as using data from real contracts. The software developed during this research is under preparation to be integrated in companies "Baltic Software Solutions" products.

1. ĮVADAS

1.1. Probleminė sritis

Spartėjant įvairiausių procesų gyvenime tempui, vis dažniau pasitaiko situacijų, kai sprendimus tenka priimti stingant informacijos, kai situacijos nepakankamai aiškiai ir visiškai apibrėžtos, ir kai priimant sprendimus dalyvauja žmogiškasis faktorius. Gyvenime dažnai kyla miglotųjų situacijų įvertinimo pagal tam tikrus parametrus problema. Situacijos gali būti vaizdai, simboliai, signalai ir t.t. Parametrai, apibūdinantys situacijas, būna ne tik skaitmeniniai, bet ir verbaliniai, pavyzdžiui, objekto savybės nusakomos žodžiais „sunkus, mažas, greitai juda“ ir panašiai. Iš tokio apibūdinimo sunku ką nors formaliai nuspręsti apie objektą, nes egzistuoja tam tikros matematinio neapibrėžtumo sąlygos ir sąvokos.

Kredito kompanijoms taip pat kyla ši problema klasifikuojant, arba atpažįstant kontraktus. Kasdien joms tenka apdoroti vis daugiau duomenų ir įvertinti būsimus kontraktus. Žmogui sudėtinga įvertinti situacijas, kurias apibūdina verbaliniai parametrai, todėl šis procesas automatizuojamas – daugelis kompanijų yra susikūrusios tam tikras kontraktų vertinimo sistemas, kurios padeda atpažinti situacijas. Šiose sistemose, dažniausiai iš jau pasibaigusių kontraktų, statistiškai apskaičiuojamas kiekvieno sutarties parametro reikšmingumo svoris, arba tą svorį numano verslo ekspertai. Paprastai daugumoje tokių sistemų statistiškai išskaičiuojant parametro svorį, vienu metu vertinamas tik vienas parametras, bet ne jo ryšys su kitais parametrais; be to, formaliai neįvertinami panašumai bei skirtumai tarp sėkmingų ir nesėkmingų kontraktų. Taip pat verslo ekspertai, subjektyviai vertindami parametro svorį, gali klysti. Todėl kyla poreikis sukurti intelektualią sistemą, kurią būtų lengva integruoti bet kurioje kompanijoje, ir kuri iš kontraktų apdorojimo istorijos ir ypač iš pasibaigusių kontraktų išskaičiuotų parametrų svorius, įvertindama ir jų tarpusavio sąryšius, ir panašumus bei skirtumus tarp sėkmingų ir nesėkmingų kontraktų. Tokia intelektuali sistema galėtų veikti kaip eksperto konsultantas ar patarėjas.

Kuriant miglotai apibrėžtos situacijos įvertinimo modulį buvo susipažinta su:

1) kreditavimo organizavimo pagrindais,
2) miglotųjų situacijų aprašymo būdais,
3) miglotųjų sistemų sudarymo pagrindais,
4) miglotojo klasterizavimo pagrindais,
5) atpažinimo ir klasifikavimo pagrindais.

Kadangi miglotai apibrėžtos situacijos įvertinimo modulis realiai kuriamas kompanijai „Baltic Software Solutions“, todėl kūrimo procesas vykdomas pagal tam tikras nustatytas darbo taisykles. Modulis realizuojamas C#.NET programavimo kalba, duomenų bazių valdymo sistema – MS SQL Server 2000. Kadangi kompanijoje projektavimas vykdomas anglų kalba, todėl šio darbo kai kuriose schemose yra angliškų terminų, kurie yra paaiškinti gimtąja lietuvių kalba.

1.2. Bendrasis tikslas ir uždaviniai

Tikslas – sukurti miglotosios situacijos įvertinimo modulį, kurį būtų lengva integruoti bet kurioje su kreditavimo procesais susijusioje kompanijoje, ir kuris iš kontraktų apdorojimo istorijos apskaičiuotų parametrų koeficientus, naudojamus naujų kontraktų atpažinimui.

Tikslui pasiekti reikia išspręsti tokius uždavinius:

1) apžvelgti esamus klasifikavimo metodus;
2) sukurti intelektualesnį, ne taisyklėmis paremtą, kontraktų atpažinimo modulį;
3) panaudoti miglotąjį klasifikavimą moduliui kurti;
4) sudaryti ir realizuoti taikymo sričiai (patyriminės istorijos kontraktams) tinkamą klasifikavimą;
5) sudaryti miglotai aprašytos situacijos įvertinimo modulio funkcinę organizaciją;
6) ištirti sudarytą modulį su:
    a) atsitiktinai sugeneruotais duomenimis,
    b) realiais patyriminių kontraktų duomenimis.

2. KLASIFIKAVIMO METODŲ APŽVALGA

2.1. Miglotosios situacijos

Gyvenime dažnai pasitaiko situacijų, kurias apibūdina ne tik deterministiniai parametrai, bet ir verbaliniai, pavyzdžiui, „protingas žmogus“, „didelis namas“ ar „šiltas oras“. Tokias situacijas įvertinti formaliai sudėtinga, nes tai daryti reikia tam tikromis neapibrėžtumo sąlygomis. Kiekvienas žmogus savaip įsivaizduoja didelį namą ar šiltą orą.

Tokios problemos kyla ir kredito kompanijoms. Jos kasdien gauna vis daugiau kontraktų pasiūlymų, kuriuos reikia įvertinti ir numatyti, ar verta kontraktą pradėti, ar ne. Kontraktus apibūdinantys parametrai būna pateikti tiek skaitine, tiek ir verbaline forma. Skaitiniai parametrai gali būti kontrakto suma, asmens pajamos ir panašiai, o verbaliniai – asmens pilietybė arba kompanijos verslo sėkmingumo mastas ir panašiai. Su skaitiniais parametrais dirbti paprasčiau, nes kiekvienas savo praktikoje susiję̨s su ta pačia skaičiavimo sistema. Asmens pajamos yra aiškus parametras, pavyzdžiui, du tūkstančiai litų yra nei daugiau, nei mažiau, o du tūkstančiai litų. Verbaliniai parametrai suteikia neapibrėžtumo, pavyzdžiui, „kliento kreditingumas yra geras“. Apibūdinimą „geras“ kiekvienas žmogus traktuoja savaip, priklausomai nuo savo patirties ir situacijos.

Dabar daugelis kredito kompanijų susikūrusios tam tikras kontraktų vertinimo sistemas. Šios sistemos sudarytos taip, kad vienu metu vertinamas tik vienas parametras, pavyzdžiui, jo statistinis vidurkis, dispersija ir panašiai. Problema ta, kad taip skaičiuojant parametrų svorį, neatsižvelgiama į visų parametrų tarpusavio ryšius, neįvertinami panašumai bei skirtumai tarp sėkmingų ir nesėkmingų kontraktų. Todėl atsiranda poreikis sukurti intelektualios sistemos modulį, kuri būtų lengva integruoti bet kurioje kompanijoje, ir kuris iš pasibaigusių kontraktų išskaičiuotų parametrų svorius, įvertindamas ir jų tarpusavio sąryšius, ir panašumus bei skirtumus tarp sėkmingų ir nesėkmingų kontraktų. Tokia sistema turėtų būti kaip kredito kompanijų patarėjas ar vertintojas bei turėtų klasifikuoti kontraktus į jiems labiausiai panašias klases. Tam pirmiausiai reikia nustatyti parametrų reikšmingumo svorius – koeficientus.

Norint apskaičiuoti koeficientus, reikia patyriminių situacijų, tai yra, kontraktų, kurie jau baigėsi, ir apie kuriuos galima pasakyti, ar tai blogas kontraktas, ar vidutinis, ar geras. Šio proceso metu gali kilti įvairių problemų. Patyriminių kontraktų gali būti labai daug, todėl gali tekti suformuluoti aukštos eilės sudėtingą uždavinį, kurio sprendimas ilgai užtruktų. Taip pat kontraktai iš vienos klasės, pavyzdžiui, geri kontraktai, gali būti per daug skirtingi, arba kontraktai iš skirtingų klasių gali būti per daug panašūs, kad primityvus koeficientų radimas pavyktų. Norint sumažinti vertinamųjų kontraktų skaičiaus įtaką, naudinga atlikti dar vieną procesą – klasterizavimą. Plačiau apie tai žr. 2.6 skyriuje.

2.2. Miglotosios aibės

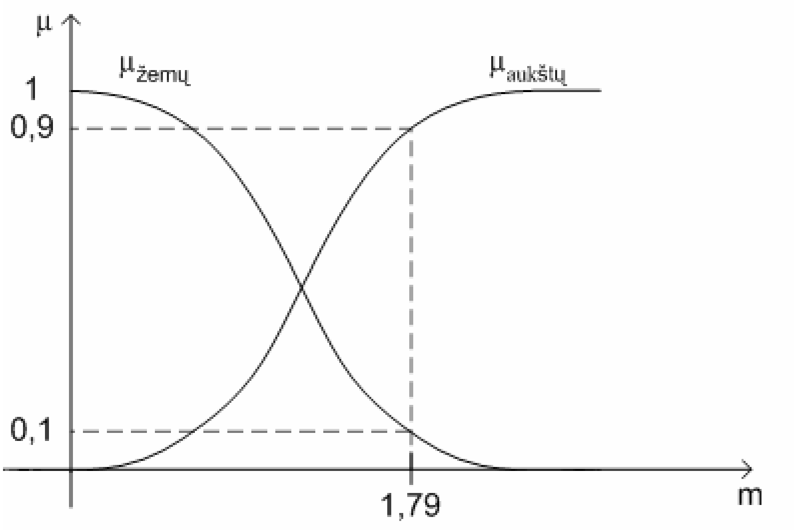
Paprastai miglotųjų situacijų aprašymas remiasi miglotųjų aibių teorinėmis sąvokomis. Kaip žinia, klasikinę deterministinę aibę patogu įsivaizduoti kaip objektų rinkinį [1]. Paprastai objektai arba priklauso tam tikrai aibei, arba ne, tai yra, riba tarp elemento buvimo aibėje ir nebuvimo yra griežta. Klasikinę aibę galima apibūdinti, taip vadinama, priklausomumo funkcija μ(x), galinčia įgauti reikšmę 1, jeigu elementas x priklauso aibei A, ir 0, jeigu elementas x nepriklauso aibei A:

$$\mu_A(x) = \begin{cases} 1, \text{jei x} \in \text{A} \\ 0, \text{jei x} \notin \text{A}. \end{cases} \tag{2.1}$$

Miglotoji aibė tarsi išplečia klasikinės aibės sampratą. Objekto priklausymo aibei tikrumas išreiškiamas tam tikra funkcija. Miglotąją aibę galima apibrėžti kaip objektų ir tų objektų priklausomumo funkcijų toje aibėje rinkinį [2]**:**

$$A = \{x, \mu_A(x)\} \tag{2.2}$$

Čia $\mu_A(\cdot)$ – aibei A priklausymo tikrumo funkcija, o $\mu_A(x)$ - elemento x priklausomumo aibei A laipsnis. Elemento x priklausomumo funkcija gali įgyti ne tik diskretines reikšmes, tai yra, 0 arba 1, bet ir visas realias reikšmes iš intervalo [0, 1]. Šiuo atveju nėra tos griežtos ribos, kuri nubrėžia liniją tarp elemento buvimo ir nebuvimo aibėje. Elemento priklausymas aibei nusakomas tam tikru skaičiumi, tai reiškia, kad jis nurodytu skaitiniu dydžiu priklauso tai aibei. Tas pats elementas tam tikru skaitiniu dydžiu gali priklausyti ir kitai aibei. Pavyzdžiui, analizuojant žmogaus ūgį klasikinės aibės atveju būtų užbrėžta tam tikra riba, kurią viršijęs žmogus – aukštas [3]. Tarkim tai būtų 1,80 metro riba. Tačiau šiuo atveju žmogus, kurio aukštis 1,79 metro, nebepriklausytų aukštų žmonių aibei. Miglotųjų aibių atveju žmogaus ūgis būtų įvertinamas tam tikru dydžiu iš pasirinkto intervalo, ir 1,79 bei 1,80 metro įverčiai turėtų atitikmenis miglotoje aibėje. Tarp jų būtų tam tikras skirtumas, tačiau ne toks griežtas kaip klasikinės aibės atveju. Išskiriant dar vieną aibę – mažų žmonių aibę, 1,79 metro bei bet kuris kitas žmogaus ūgis turėtų atitikmenį ir toje aibėje, tik mažesnį, pavyzdžiui, ūgis 1,79 metro aukštų žmonių aibėje galėtų būti įvertintas 0,9, o mažų žmonių aibėje – 0,1 (2.1 pav.).



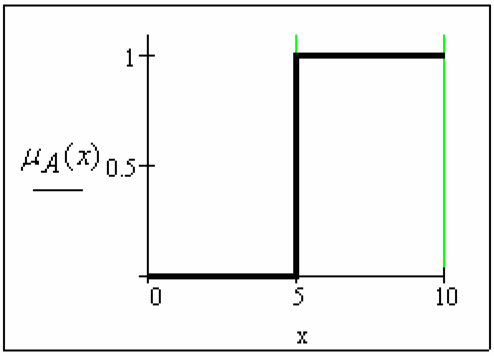
2.1 pav. Žmogaus ūgio priklausomumo funkcijos

Norint išryškinti skirtumą tarp klasikinės ir miglotosios aibės, galima paimti po vieną šių aibių pavyzdį:

1) didesnių arba lygių penkiems realių skaičių aibė A,

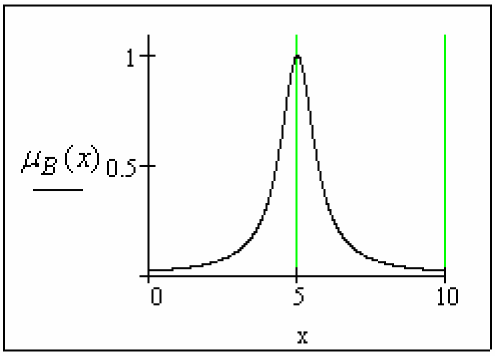
2) artimų penkiems realių skaičių aibė B.

Pirmoji aibė A yra klasikinė ir turi griežtą ribą tarp aibei priklausančių ir nepriklausančių objektų (2.2 pav.). Iš grafiko matyti, jog visų skaičių, nors šiek tiek mažesnių nei 5, priklausomumo aibei funkcijos $\mu_A(x)$ reikšmė – 0. Vadinasi tie skaičiai aibei A nepriklauso. Visų kitų realių skaičių, pradedant 5 ir nors kiek didesnių nei 5, priklausomumo aibei funkcijos reikšmė – 1, todėl tie skaičiai aibei priklauso.



2.2 pav. Klasikinės aibės priklausomumo funkcija

Antroji aibė B yra miglotoji (2.3 pav.). Vien iš aibės apibūdinimo, jog į aibę patenka penketui artimi skaičiai, matyti, kad čia nėra tokios griežtos ribos. Artimesnių penketui skaičių priklausomumo aibei funkcijų reikšmės didesnės ir labiau priklauso aibei, tolimesnių – mažesnės, o labiausiai tolimų lygios nuliui.



2.3 pav. Miglotosios aibės priklausomumo funkcija

Aibės B apibūdinimas nevisai tiksliai apibrėžtas. Žodis „artimas“ taip pat yra miglotas, ir kiekvienas jį gali įsivaizduoti savaip. Todėl aibės priklausomumo funkciją galima skirtingai aprašyti, pavyzdžiui, kaip 2.3 ir 2.4 formulėse.

$$\mu_B(x) = \frac{1}{1+10(x-5)^2} \tag{2.3}$$

$$\mu_B(x) = \frac{1}{1+(x-5)^2} \tag{2.4}$$

Žinoma, kad miglotosios aibės viršutinė riba – tai supremumas, tai yra mažiausias realus skaičius, kuris yra didesnis arba lygus bet kuriam aibės elementui (2.5). Supremumas žymimas sup(S), kur S – aibė. Kelios miglotosios aibės laikomos normuotomis, kai jos turi vienodą supremumą, pavyzdžiui, lygų vienetui. Kitu atveju aibės laikomos nenormuotomis.

$$\begin{aligned} &\sup\{1,2,3\} = 3 \\ &\sup\{x \in R : 0 < x < 1\} = \sup\{x \in R : 0 \le x \le 1\} = 1 \end{aligned} \tag{2.5}$$

2.3. Miglotosios sistemos

Formaliam miglotųjų situacijų apdorojimui informacinėms technologijoms talkina miglotoji logika daugelyje mokslinių programų ir inžinerinių sistemų, ypač valdymo sistemose bei vaizdų atpažinime [1]. Miglotųjų sistemų naudojimas labai išplito įvairiuose vartotojų bei pramoniniuose produktuose, pavyzdžiui, skalbimo mašinose, autofokusinėse kamerose, oro kondicionieriuose, miniatiūriniuose kompiuteriuose, dulkių siurbliuose, automobilių transmisijose, laivų navigatoriuose, metro, cemento džiovinimo krosnyse ir kitose srityse [1]. Miglotoji logika taip pat naudojama ir intelektualiose sistemose. Tokios intelektualiosios sistemos turi savotiškas „mąstymo“ galimybes su minimaliais apribojimais ir pačios sugeba priimti sprendimus.

Miglotosios aibės – tai matematinis būdas, padedantis apdoroti neapibrėžtumą verbalinėse sąvokose. Jas galima vertinti kaip klasikinės aibių teorijos apibendrinimą, nes jos suteikia reikiamo lankstumo (plačiau 2.2 skyriuje). Pasak Lotfi Zadeh [2], esant neapibrėžtumo sąlygoms, miglotumą patogu įvertinti daugelyje sričių, išplečiant tradicinių aibių koncepciją iki miglotųjų aibių, pavyzdžiui, aritmetiką, grafų teoriją, tikimybių teoriją galima atitinkamai išplėsti iki miglotosios aritmetikos, miglotosios grafų teorijos ir miglotosios tikimybių teorijos. Miglotumas taip pat būdingas ir kitose srityse, pavyzdžiui, neuronų tinkluose, genetikos algoritmuose, stabilumo teorijoje, vaizdų atpažinime, matematiniame programavime [1]. Miglotumo formalizavimas suteikia didesnį bendrumą, didesnę išraiškos galimybę bei kokybiškesnę realaus pasaulio problemų modeliavimo galimybę.

Miglotumas dažnai painiojamas su tikimybe. Pagrindinis skirtumas tarp jų tas, kad miglotumas nagrinėja deterministinį tikėtinumą, o tikimybė siejasi su nedeterministinių, atsitiktinių įvykių tikėtinumu. Miglotumas – tai neapibrėžtumas, nes tam tikri terminai gali būti neaiškūs ir neapibrėžti, pavyzdžiui, „jaunas žmogus“ arba „didelis kambarys“. Tačiau tikimybė simbolizuoja atsitiktinumo idėją. Teiginys yra tikimybinis, jeigu jis yra pasekmė aiškiai apibrėžtų, bet atsitiktinių įvykių, pavyzdžiui, „mestas kauliukas išris šešis“. Taigi miglotumas ir atsitiktinumas atspindi skirtingus neapibrėžtumo aspektus. Pirmasis perteikia subjektyvų žmogaus mąstymą, jausmus ar kalbą, o antrasis – objektyvią statistiką. Iš modeliavimo pozicijų žiūrint, miglotieji ir statistiniai modeliai apdoroja filosofiškai skirtingas informacijos rūšis– pirmieji susiję su objektų panašumo atvaizdavimu į netiksliai apibūdintas savybes, o tikimybė perteikia informaciją apie susijusius pasikartojamumus [1].

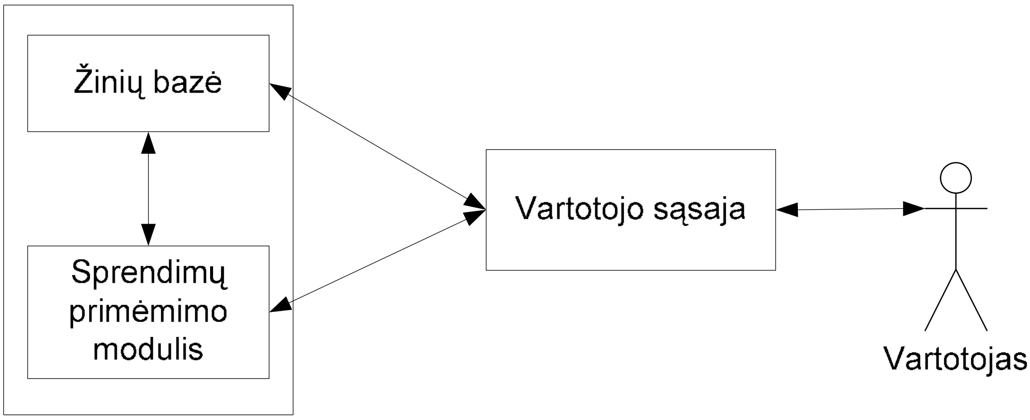
Svarbi miglotųjų situacijų vertinimo sistema – ekspertinė sistema. Tai programa, kuri „elgiasi“ kaip tam tikros probleminės srities ekspertas [1]. Tokia sistema geba spręsti uždavinius, naudodama probleminei sričiai būdingą informaciją. Iš ekspertinės sistemos tikimasi, kad ši mokėtų nagrinėti netiksliai apibrėžtą ir nepilną informaciją. Jos vartotojas gali nustatyti sistemos tikslus ir elgesį uždavinio sprendimo metu ir po jo.

Paprastai ekspertinė sistema susideda iš trijų pagrindinių dalių (2.4 pav.):

1) žinių bazės,

2) sprendimų priėmimo modulio,

3) vartotojo sąsajos.



2.4 pav. Ekspertinė sistema

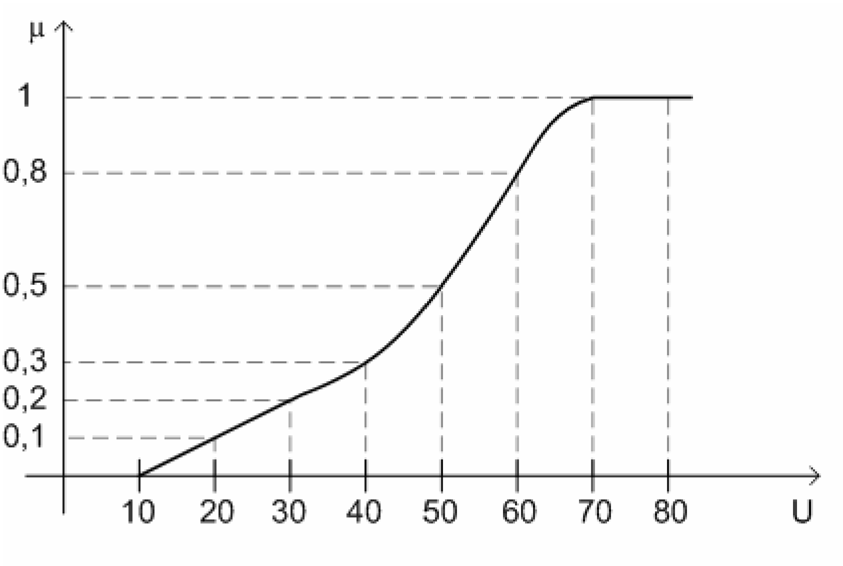
Žinių bazė apima informaciją apie probleminę sritį įskaitant faktus ir taisykles, galiojančius toje srityje. Fakto pavyzdys – „katė didesnė už pelę“. Taisyklės dažnai apibūdinamos tokiu principu „jei ..., tai ...“, pavyzdžiui, „jei žmogus rūko, tai susirgs vėžiu“. Tokie užrašymai formalizuojami pagal tam tikras taisykles. Sprendimų priėmimo modulis, naudodamasis informacija iš žinių bazės, atlieka reikiamus skaičiavimus ir grąžina rezultatus pagal vartotojo užklausas. Vartotojo sąsaja – tai tarpininkas tarp vartotojo ir sistemos.

Svarbu, kad ekspertinė sistema susidorotų su miglotumo problema. Daugumoje sistemų ši problema sprendžiama tikimybiniais metodais [4]. Pavyzdžiui, kiekviena taisyklė sistemoje turi savo stiprumo laipsnį, vadinamą tikrumo faktoriumi, kuris kinta intervale [0;1]. Kiekviena taisyklė taip pat turi savo įvykio stiprumą iš intervalo [-1;1]. Jeigu įvykio stiprumas pakankamai didelis, pavyzdžiui, stiprumo reikšmė iš intervalo [-0,2;0,2], tuomet taisyklės rezultatas gaunamas įvykio stiprumą padauginus su taisyklės tikrumo faktoriumi. Skirtingų taisyklių rezultatai sujungiami ir gaunama vartotojo iškeltos hipotezės tikėtinumo ar netikėtinumo įverčiai atitinkamai iš intervalų [0;1] ir [-1;0]. Hipotezė laikoma teisinga, jei gautas tikėtinumo įvertis didesnis už tam tikrą nustatytą slenkstį. Taip pat hipotezė laikoma neteisinga, jei netikėtinumo įvertis mažesnis už tam tikrą slenkstį.

Toks metodas pakankamai efektyvus tam tikrais atvejais. Tačiau ekspertai dažnai mąsto ne tikimybinėmis reikšmėmis, bet formaliai neapibrėžtomis žodinėmis išraiškomis, pavyzdžiui, „dažnai“, „daug“. Jeigu miglotosios aibės ir verbaliniai terminai geriausiai apibūdina probleminę sritį, tai motyvuoja panaudoti miglotąją logiką tradicinėse ekspertinėse sistemose.

2.4. Kokybinių aprašų transformavimas į kiekybinius

Ekspertams savo idėjas apie probleminę sritį paprasčiau perteikti naudojant lingvistinius kintamuosius [5]. Lingvistiniai kintamieji yra svarbūs miglotoje logikoje, ypač miglotose ekspertinėse sistemose. Iš prigimties lingvistinis kintamasis – tai kintamasis, kurio reikšmės yra žodžiai ar sakiniai natūralioje ar dirbtinėje kalboje [6]. Prieš pateikiant lingvistinio kintamojo formalų apibrėžimą, reikia formaliai apibrėžti miglotąjį kintamąjį. Miglotasis kintamasis charakterizuojamas (X, U, R(X)), kur X– kintamojo vardas, U– reikšmių aibė, o R(X)– miglotasis aibės U poaibis, kuris parodo kintamojo X miglotąjį apribojimą. Tarkim X= „Senas“, o U={10, 20,..., 80}, tuomet R(X)= {0.1/20; 0.2/30; 0.3/40; 0.5/50; 0.8/60; 1/70; 1/80} yra kintamojo „Senas“ miglotasis apribojimas (2.5 pav.) [1].



2.5 pav. Kintamojo „Senas“ priklausomumo funkcija

Lingvistinis kintamasis yra aukštesnio rango nei miglotasis kintamasis [1]. Miglotieji kintamieji yra lingvistinio kintamojo reikšmės. Lingvistinis kintamasis charakterizuojamas (x, T(x), U, G, M), kur x– kintamojo vardas, T(x)– kintamojo x reikšmių aibė, kur kiekviena reikšmė– tai miglotasis kintamasis, apibrėžtas aibėje U, G– sintaksės taisyklė, kaip generuoti kintamojo x reikšmių vardus, o M– semantinė taisyklė, kaip susieti kiekvieną x reikšmę su tam tikra verte [1]. Pavyzdžiui, lingvistinis kintamasis x = „greitis“, U = [0; 100]. Tuomet pagal sintaksės taisyklę G sugeneruojami x reikšmių vardai ir T(greitis) = {Labai lėtas, Lėtas, Vidutinis, Greitas,...}. Semantinė taisyklė M galėtų būti atvaizduota taip:

M(Lėtas) = miglota aibė „greičio, mažesnio už apie 40 km/h“,

kurios priklausomumo funkcija – μ_{Letas}.

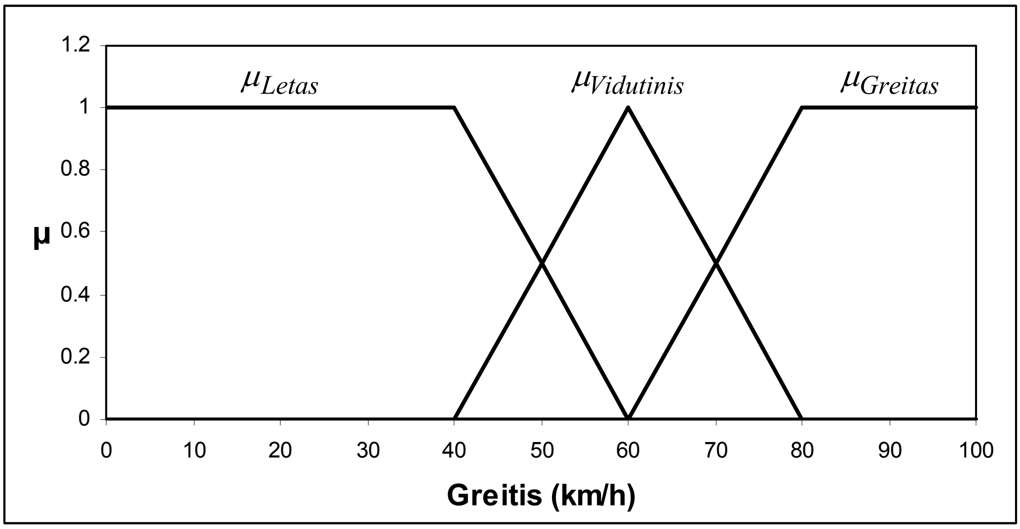
M(Vidutinis) = miglota aibė „greičio apie 60 km/h“,

kurios priklausomumo funkcija – $\mu_{Vidutinis}$.

M(Greitas) = miglota aibė „greičio didesnio už apie 80 km/h“,

kurios priklausomumo funkcija – $\mu_{Greitas}$.

Šių priklausomumo funkcijų grafikas 2.6 paveiksle.



2.6 pav. Greičio priklausomumo funkcijos

2.5. Atpažinimo ir klasifikavimo metodai

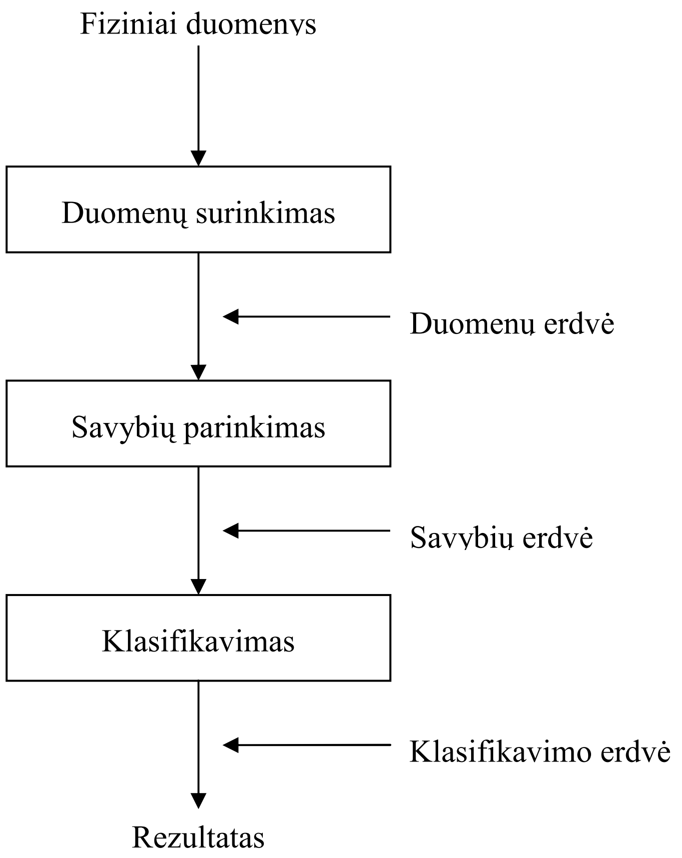
Iš principo žinomi du situacijų, vaizdų ar modelių atpažinimo būdai. Pirmasis – tai atpažinimas pagal instrukcijas, kurias pateikia ekspertas ar mokytojas. Šis būdas tinka tik paprastų situacijų ar vaizdų atpažinimui. Antrasis – tai atpažinimas pagal principą „daryk kaip aš" [1]. Tai yra, ekspertas pats nežino, kodėl jis vieną ar kitą situaciją priskiria konkrečiai klasei, bet žino sėkmingų priskyrimų atvejus (pozityvinę patyriminę istoriją) ar nesėkmingų priskyrimų atvejus (negatyvinę patyriminę istoriją). Šis būdas naudojamas ekspertinių sistemų apmokymui. Naudojantis miglotųjų aibių teorija, buvo sukurta keletas atpažinimo metodų, kurie sėkmingai buvo realizuoti įvairiose srityse: kalbos, simbolių, geometrinių objektų atpažinime, intelektualių robotų kūrime, vaizdų apdorojime, signalų klasifikavime bei programose, skirtose medicinai.

Modelių atpažinimas paprastai susideda iš trijų žingsnių [7] (2.7 pav.):

1) duomenų surinkimo,

2) savybių parinkimo,

3) klasifikavimo.



2.7 pav. Modelių atpažinimo procesas

Pirmiausia surenkami klasifikavimui reikalingi duomenys, kurių gali būti labai daug. Jie gali būti skaitmeninėje ir/ar lingvistinėje formoje. Duomenų kiekį reikia minimizuoti, todėl atliekamas savybių parinkimas iš turimų duomenų. Pagal parinktas savybes vėliau atliekamas klasifikavimas. Pageidaujama, jog savybių erdvės dimensijos būtų daug mažesnės negu duomenų erdvės dimensijos, kad klasifikavimas būtų efektyviai pritaikytas ir praktiškai suvoktas. Klasifikavimą atlieka klasifikatorius, kuris yra tarsi transformacija tarp savybių ir klasių. Objekto aprašas, kurį apibūdina parinktos savybės, pateikiamas klasifikatoriui, ir jis priskiriamas tai klasei, į kurią yra labiausiai panašus.

Miglotųjų aibių koncepcija šiame procese randama dviejose vietose:

1) savybių erdvėje,

2) klasifikavimo erdvėje.

Suprantama, jog dauguma informacijos, surinktos per realių situacijų atpažinimo procesus, yra ne skaitmeninio tipo. Taip pat žmogus, apibendrindamas skaitmeninius duomenis, gali juos paversti verbaliniais. Todėl, pirma, miglotosios aibės savybių erdvėje padeda atvaizduoti lingvistinius apibūdinimus. Antra, skirtingai nei griežtai klasifikuojant, kai atpažįstamas objektas priklauso tik vienai klasei, miglotai klasifikuojant tas objektas gali priklausyti skirtingom klasėm, tačiau skirtingais priklausymo (panašumo) laipsniais. Pavyzdžiui, objektas gali būti panašus į vieną klasę santykiniu dydžiu 0,8, o į kitą klasę – 0,4, todėl paprastai jam priskiriama ta klasė, į kurią labiausiai panašus, tai yra, panašumo laipsnis yra didžiausias. Tačiau gali būti, kad objektas vienodai arba labai panašus į vieną ar kelias skirtingas klases. Todėl miglotasis klasifikavimas suteikia lankstumo, nes galima įvairiai sukonfigūruoti klasifikatorių; tarkim, objektą priskirti vienai klasei, kai jo panašumo laipsnis į tą klasę yra ne mažiau kaip 15% didesnis negu į kitas klases.

Miglotumas ir tikimybė susiję su skirtingais neapibrėžtumo aspektais objektų atpažinimo procesuose. Skirtumą tarp miglotumo ir tikimybės galima paryškinti šiais dviem teiginiais:

1) objekto priklausomumo klasei ωi laipsnis yra α, kur $\alpha \in [0;1]$,

2) tikimybė, kad objektas priklauso klasei ωi yra α, kur $\alpha \in [0;1]$.

Pirmajame teiginyje α parodo objekto panašumo laipsnį į klasę ω_i. Antrasis teiginys reiškia, kad, jeigu objektas su tuo pačiu savybių įverčių rinkiniu pasitaiko N kartų, tai αN kartų jis priklauso klasei ω_i.

Statistiniai (tikimybiniai) metodai turi ilgą istoriją atpažinimo technologijų kūrime. Jeigu surinktų duomenų kiekis mažas, tai tikimybiniais metodais neįmanoma pasikliaut. Taip pat, jei dalis duomenų pateikta lingvistine forma, tai miglotasis klasifikavimas – geras pagalbininkas.

2.6. Miglotasis klasterizavimas

Klasterizavimas – tai klasės objektų išskaidymas į tam tikrą kiekį daugiau ar mažiau panašių poklasių (klasterių) taip, kad objektai, priklausantys vienam klasteriui, būtų panašūs, o objektai iš skirtingų klasterių skirtųsi kiek įmanoma daugiau [1]. Klasterizavimą naudinga daryti tada, kai apdorojamų duomenų yra daug, arba kai ta pati klasė apima pakankamai skirtingus objektus. Nemiglotame klasterizavime ribos tarp klasterių griežtos, tai yra, objektas gali priklausyti tik vienam klasteriui. Miglotojo klasterizavimo atveju, objektas gali priklausyti daugiau nei vienam klasteriui, tačiau skirtingais priklausomumo (panašumo) laipsniais. Objektų panašumas į klasterius pagrįstas tokiu pačiu principu kaip ir objektų panašumas į klases, nes klasteris iš esmės ir yra klasė. Skirtumas tarp klasės ir klasterio tik toks, kad klasė yra aukštesnio lygio objektų aibė, ir klasė gali būti sudaryta iš tam tikro skaičiaus klasterių. Paprastai laikoma, kad objektas priklauso tai klasei, į kurios klasterį jis labiausiai panašus.

Literatūroje žinomas ne vienas miglotojo klasterizavimo algoritmas, pavyzdžiui, miglotasis c-klasterių (angl. *fuzzy c-means)*, artimiausio kaimyno (angl. *Nearest neighborhood*) [1, 5, 8]. Pagrindinė klasterizavimo algoritmų bėda ta, kad jie nevertina skirtumų tarp situacijų iš skirtingų klasių. Taip pat, prieš pradedant klasterizuoti, iš anksto nustatomas būsimų klasterių kiekis. Tai ne visada gerai, nes, prieš sprendžiant uždavinį, gali būti neaišku, koks yra objektų pasiskirstymas klasėje; tai yra, ar objektai labai panašūs vieni į kitus, ar skirtingi. Pavyzdžiui, jeigu situacijos apytikriai pasiskirsčiusios į tam tikrą grupių kiekį, tai geriau tas situacijas apjungti į tokį patį kiekį klasterių, o ne mažesnį ar didesnį.

Artimiausio kaimyno algoritmas situacijas nagrinėja pagal atstumo matą. Panašumo matas natūralesnis, nes, norint iškelti hipotezę, kokiais bruožais situacijos panašios arba skiriasi, tai paprasčiau pasakyti, kad situacijos tam tikrais parametrais panašios arba nepanašios, o ne teigti, jog situacijos artimos arba tolimos.

2.6.1. Miglotojo klasterizavimo formalizavimas

Tarkim, kad yra baigtinė objektų aibė $X = \{x_1, x_2, ..., x_n\}$, kurios elementai priklauso p-matei Euklido erdvei R^P ($x_j \in R^P$, j=1,2,...,N). Užduotis – pagal tam tikrą kriterijų išskaidyti šios aibės objektus į klasterių kiekį, kurį nusako skaičius c. Miglotojo klasterizavimo rezultatą galima išreikšti priklausomumo matrica U [1]:

$$U = [u_{ij}]_{i=1...c,\, j=1...N}\ . \tag{2.6}$$

Čia u_{ij} išreiškia laipsnį, kuriuo elementas x_j priklauso i-tajam klasteriui ir gali įgyti reikšmę iš intervalo [0;1]. Reikšmė u_{ij} turi dar du papildomus apribojimus. Pirma, elemento x_j priklausomumo laipsnių visuose klasteriuose suma lygi vienam:

$$\sum_{i=1}^{c} u_{ij} \le N\ , \text{ j=1,2,...,N.} \tag{2.7}$$

Antra, kiekvienas klasteris nėra tuščias, bet ir mažesnis už aibę X:

$$0 < \sum_{j=1}^{N} u_{ij} < n\ , \text{ i=1,2,...,c.} \tag{2.8}$$

Bendra objektinė funkcija atrodo taip [1]:

$$J(u_{ij}, v_k) = \sum_{i=1}^{c} \sum_{j=1}^{N} \sum_{k=1}^{c} g[w(x_i), u_{ij}]\, d(x_j, v_k)\ ; \tag{2.9}$$

čia $w(x_i)$– aprioriniai kiekvieno x_i svoriai, o $d(x_j,v_k)$– nepanašumo laipsnis tarp x_j ir v_k, kuris yra centrinis k-tojo klasterio vektorius. Nepanašumo laipsnis tenkina dvi aksiomas:

$$\begin{cases} d(x_j, v_k) \ge 0 \\ d(x_j, v_k) = d(v_k x_j) \end{cases} \tag{2.10}$$

Tuomet miglotąjį klasterizavimą galima suformuluoti kaip optimizavimo uždavinį [1]:

$$J(u_{ij}, v_k) \to \min \qquad \text{, i, k= 1,2,...,c; j=1,2,...,N}$$

esant (2.7) ir (2.8) apribojimams

3. ATPAŽINIMO METODO TEORINĖ REALIZACIJA

3.1. Parametrų norminimas

Situacijų (kontraktų) bei jos savybių (parametrų) įverčių palyginimui tarpusavyje, situacijas būtina norminti. Kiekviena situacija aprašoma savybių įverčių rinkiniu:

$$\{C_1,...,C_i,...,C_N\}. \qquad (3.1)$$

Savybės įvertis C_i gali būti trijų tipų:

1) skaitinis (20),
2) verbalinis (daug, didelis),
3) nereikšminis (mėlynas, arklys).

Paprastai kiekvienas C_i turi savo baigtines ribas (3.2). Pagal jas ir galima atlikti norminimą (3.3).

$$C_i \in [C_{i\,\min}, C_{i\,\max}] \quad \forall_i \qquad (3.2)$$

Tarkim, turint situacijos parametrą C_1 „mėnesinė alga“, kur C_{1min}=0 ir C_{1max}=5000. Taip pat situacijos formalizavimui būtina, kad jos įverčiai būtų išmatuojami ir palyginami. Dėl to, atliekant norminimą pagal (3.3), ir priėmus, kad C=10, gaunasi, jog A_{1min}=0 ir A_{1max}=10, o 1000 dydžio atlyginimas atitinka A_1=2.

$$\begin{cases} C_{i\min} = 0, \quad C_{i\max} = C \quad \forall_i \\ \text{Atitiktis} \quad C_i \leftrightarrow A_i \\ A_i \in [0, C] \end{cases} \qquad (3.3)$$

Verbaliniams įverčiams ekspertas turi suteikti skaitinius įverčius iš to paties intervalo [0, C]. Vienodai verbalinio įverčio sąvokai turi atitikti vienodas skaitinis įvertis, tai yra, jeigu tam tikro parametro verbalinis įvertis „vidutinis“ turi skaitinį atitikmenį 3, tai visų situacijų to parametro verbalinis įvertis „vidutinis“ turi būti transformuojamas į 3. Nereikšminiams įverčiams, jeigu ekspertas apie jų gerumą nieko negali pasakyt, pavyzdžiui, ar „juodas“ geriau už „mėlyną“, priskiriami atsitiktiniai skaičiai taip pat iš to paties intervalo [0, C]. Po norminimo gaunamas savybių įverčių rinkinys, arba situacijos aprašas:

$$\{A_1,...,A_i,...,A_N\} \qquad (3.4)$$

3.1.1. Parametrų norminimo pavyzdys

Tegul situacija S aprašyta tokiu savybių įverčių rinkiniu (3.1 lentelė).

3.1 lentelė. Situacijos S savybių įverčiai

C_1	C_2	C_3	C_4
17	Prastas	Arklys	10%

Savybės turi savo baigtines ribas. Visos savybės norminamos į intervalą [0;10]. Verbalinio parametro C_2 galimiems įverčiams parenkami adekvatūs prasmei atitikmenys, o nereikšminio parametro C_3, apie kurio elementų reikšmingumą ekspertai neturi jokios nuomonės, įverčiams atitikmenys parenkami atsitiktinai (3.2 lentelė).

3.2 lentelė. Savybių įverčių norminimai

C_1 = [10-20]	→ [0-10]
C_2 = [Prastas, Vidutinis, Geras, Puikus]	→ [0, 3, 6, 10]
C_3 = [Asilas, Kupranugaris, Arklys]	→ [0, 5, 10]
C_4 = [0 % - 100%]	→ [0-10]

Pagal 3.2 lentelę sunorminus situacijos S savybių įverčius, nurodytus 3.1 lentelėje, gaunamas sunormintų savybių įverčių rinkinys (3.3 lentelė).

3.3 lentelė. Situacijos S sunormintų savybių įverčiai

A_1	A_2	A_3	A_4
7	0	10	1

3.2. Miglotųjų situacijų centravimas

Dažnai praktikoje yra naudingas miglotųjų situacijų centravimas, nes išryškina savybių įverčių nukrypimus nuo vidurkio. Savybių reikšmingumo koeficientų skaičiavimo uždaviniuose (3.3.1 skyrius) lyginant tarpusavyje situacijas tai padeda išryškinti koeficientus, kurie būdingesni vienai ar kitai klasei. Miglotųjų situacijų centravimas gali būti atliekamas tik po norminimo. Kadangi po norminimo miglotoji situacija aprašoma (3.4) rinkiniu, tai centravimą sudaro savybių įverčių vidurkio atėmimas iš kiekvieno situacijos savybės įverčio:

$$a_i = A_i - \frac{1}{N}\sum_{j=1}^{N} A_j, \quad \forall_i \tag{3.5}$$

Po centravimo gaunamas naujas savybių įverčių rinkinys:

$$\{a_1, ..., a_i, ..., a_N\} \tag{3.6}$$

Centravimo savybės:

1) situacijos visų savybių įverčių suma lygi nuliui, ir dėl to, formuojant reikšmingumo koeficientų radimo uždavinį, paprasčiau atlikti duomenų patikrinimą:

$$\sum_{i=1}^{N} a_i = 0 \tag{3.7}$$

2) dirbama su „kontrastais“; tarpusavyje lyginant situacijas išryškinamas ne parametrų įverčių amplitudinis dydis, o pačių situacijų aprašų fliuktacijų skirtumai ar panašumai.

Jeigu centravimas atliekamas sunormuotai miglotajai situacijai, aprašytai 3.3 lentelėje, tai jos vidurkis – 4.5. Po centravimo rezultatas pavaizduotas 3.4 lentelėje.

3.4 lentelė. Situacijos S sucentruotų savybių įverčiai

a_1	a_2	a_3	a_4
2.5	-4.5	5.5	-3.5

Centruojant kitą situaciją S_2 (3.5 lentelė), rezultatas 3.6 lentelėje.

3.5 lentelė. Situacijos S_2 sunormintų savybių įverčiai

A_1	A_2	A_3	A_4
7	0	3	2

3.6 lentelė. Situacijos S_2 sucentruotų savybių įverčiai

a_1	a_2	a_3	a_4
4	-3	0	-2

Abiejose dar necentruotose situacijose pirmasis parametras turi įvertį 7. Po centravimo įverčiai jau turi skirtingas reikšmes – antrosios situacijos a_1 parametro įvertis didesnis, nes antrojoje situacijoje šis parametras dominuoja, todėl yra daugiau išryškintas nei pirmoje situacijoje.

3.3. Atpažinimo uždavinio formulavimas

3.3.1. Savybių reikšmingumo koeficientų radimo uždavinio formulavimas

Tegul p-tosios klasės lp-toji situacija aprašoma tokiu parametrų įverčių rinkiniu:

$$S_{lp}^{p} = (a_{lp,1}^{p},...,a_{lp,i}^{p},...,a_{lp,N}^{p})\text{; lp=1,2,...,Lp} \tag{3.8}$$

Reikia apskaičiuoti p-tajai klasei parametrų koeficientus, pagal kuriuos vykdomas atpažinimas:

$$K^{p} = (k_{1}^{p},...,k_{i}^{p},...,k_{N}^{p}) \tag{3.9}$$

Koeficientuose turi būti akumuliuota tai, kas bendra nagrinėjamai patyriminės istorijos klasei, ir tai, kas skiria ją nuo kitų klasių, tai yra, koeficientų rinkinys (3.9), kaip atstovas p-tajai klasei, turi būti kuo panašesnis į p-tosios klasės situacijų parametrų įverčius ir skirtis nuo visų kitų klasių situacijų parametrų įverčių. Dėl to įvedama speciali panašumo mato funkcija $\Phi(S^{p}, K^{p})$, kuri privalo turėti kuo didesnę reikšmę visoms p-tosios klasės situacijoms ir kuo mažesnę reikšmę kitų klasių situacijoms [9]. Tokiu būdu galima suformuluoti po uždavinį kiekvienai klasei. Tam reikia maksimizuoti panašumo mato funkciją $\Phi(S^{p}, K^{p})$:

$$G(K^p) = \max \Phi(S_{kp}^p, K^p) \qquad (3.10)$$

esant tokiems apribojimams:

$$\begin{cases} \Phi(S_{lp}^p, K^p) \geq \gamma \cdot \Phi(S_{kp}^p, K^p); & \mathrm{lp} = 1,\ldots,\mathrm{Lp} \neq \mathrm{kp} \\ \Phi(S_{lp}^r, K^p) \leq \kappa \cdot \Phi(S_{kp}^p, K^p); & \mathrm{lr} = 1,\ldots,\mathrm{Lr}; \quad \mathrm{r} = 1,\ldots,\mathrm{Z} \neq \mathrm{p} \\ 0 \leq k_i^p \leq D; \forall i \\ \gamma > 0, \gamma > \kappa. \end{cases} \qquad (3.11)$$

Čia $G(K^p)$ – p-tosios klasės maksimizuojama funkcija, sudaryta pagal atstovaujančią klasei situaciją S_{kp}^p; $\Phi(S_{lp}^p, K^p)$ – funkcija, sudaryta pagal situaciją S_{lp}^p; Z – klasių skaičius; Lp – p-tajai klasei priklausančių situacijų skaičius; Lr – r-tajai klasei priklausančių situacijų skaičius; N – bet kurios iš Z istorinių klasių situacijos parametrų skaičius; k_i^p – p-tosios klasės i-tojo parametro koeficientas; $a_{lp,i}^p$ – p-tosios klasės lp-tosios situacijos i-tasis parametras; γ – dydis, skirtas p-tosios klasės atstovaujančiai situacijai palyginti su savo klasės likusiais atstovais; κ – dydis, skirtas p-tosios klasės atstovaujančiai situacijai palyginti su visų kitų klasių atstovais; D – dydis, parodantis, kokią maksimalią reikšmę gali įgauti koeficientai.

Formulėje 4.12 $\mathrm{lp} \neq \mathrm{kp}$, kad nereikėtų lyginti situacijos panašumo pačios su savimi, o $\mathrm{r} \neq \mathrm{p}$, kad būtų lyginamas tik p-tosios klasės atstovaujančios situacijos nepanašumas su kitų klasių situacijomis. Kiekvienai klasei formuojamas uždavinys su tokiomis pačiomis γ, κ, ir D reikšmėmis. Apribojimas D atlieka ir norminimo funkciją, nes visoms klasėms koeficientai gaunami tose pačiose ribose. Kad visų klasių maksimizuojamos funkcijos būtų palyginamos, jas reikia norminti, tai yra, apskaičiuoti kiekvienai klasei konstantas c^p (p=1,...Z):

$$c^1 \cdot G(K^1) = \ldots = c^2 \cdot G(K^2) = \ldots = c^p \cdot G(K^p) \qquad (3.12)$$

Toliau reikia parinkti panašumo mato funkcijos tipą. Čia kyla dvi prieštaraujančios tendencijos:

1) kuo galingesnė ir sudėtingesnė funkcija $\Phi(S^p, K^p)$, tuo tiksliau galima atskirti klases ir atpažinti situacijas su mažesne klaidų tikimybe;
2) kuo paprastesnė panašumo funkcija, tuo mažiau resursų reikia koeficientų radimui bei situacijų klasifikavimui.

Literatūroje [9] žinomos atpažinimo sistemos naudoja įvairias panašumo funkcijas: tiesines, kvadratines, gabalais tiesines ir panašiai. Panašumo mato funkciją pasirinkus paprasta tiesine arba gabalais tiesine funkcija, 3.10 ir 3.11 formulės atitinkamai susiveda į tiesinio arba gabalais tiesinio programavimo uždavinį. Bendru atveju patogu pasirinkti gabalais tiesinę funkciją:

$$\Phi(S_{kp}^{p}, K^{p}) = \max_{wp=1,\ldots,Wp} \sum_{i=1}^{N} a_{kp,i}^{p} \cdot k_{wp,i}^{p} \,. \tag{3.13}$$

Čia Wp – tai p-tosios klasės poklasių (klasterių) skaičius; $a_{kp,i}^{p}$ – p-tosios klasės kp-tosios situacijos i-tojo parametro įvertis; $k_{wp,i}^{p}$ – p-tosios klasės wp-tojo klasterio i-tojo parametro koeficientas.

Taip suformuluojamas gabalais tiesinio programavimo uždavinys, kuris atrodo taip:

$$\max\ (3.13) > 0$$

$$\textit{esant tokiems apribojima ms}$$

$$\begin{cases} \max\limits_{\mathrm{W}} (S_{lp}^{p} \cdot K_{w}^{p}) - \gamma \max\limits_{W} \sum\limits_{i=1}^{N} a_{kp,i}^{p} \cdot k_{w,i}^{p} \geq 0; \forall \mathrm{lp} \\ \max\limits_{\mathrm{W}} (S_{lr}^{r} \cdot K_{w}^{p}) - \kappa \max\limits_{W} \sum\limits_{i=1}^{N} a_{kp,i}^{p} \cdot k_{w,i}^{p} \leq 0; \forall lr; \forall r \neq p \\ 0 \leq k_{w,i}^{p} \leq D; \forall i; \forall w \end{cases} \tag{3.14}$$

Uždavinyje (3.14) reikalaujama, kad p-tosios klasės situacijų panašumo funkcijų reikšmės skirtųsi nuo p-tosios klasės atstovaujančios situacijos S_{kp}^{p} panašumo funkcijos reikšmės ne daugiau kaip per γ, ir kad bet kurios kitos klasės situacijų panašumo funkcijų reikšmės neviršytų situacijos S_{kp}^{p} panašumo funkcijos reikšmės per κ.

Norėtųsi, kad panašumo mato funkcija būtų kuo paprastesnė, bet efektyvi ir plačiai išnagrinėta matematikoje. Todėl gabalais tiesinio programavimo uždavinį galima skaidyti į keletą tiesinio programavimo uždavinių ir taip suformuluoti po tokį uždavinį kiekvienam klasės poklasiui; tai yra:

$$\Phi^p = \sum_{i=1}^{N} k_i^p \cdot a_{1,i}^p \to \max$$

$$\begin{cases} \sum_{i=1}^{N} k_i^p \cdot a_{lp,i}^p \geq \gamma \sum_{i=1}^{N} k_i^p \cdot a_{1,i}^p; \qquad \mathrm{lp} = 1,\ldots,\mathrm{Lp} \\ \sum_{i=1}^{N} k_i^p \cdot a_{lr,i}^r \leq \kappa \sum_{i=1}^{N} k_i^p \cdot a_{1,i}^r; \qquad \mathrm{lr} = 1,\ldots,\mathrm{Lr}; \qquad \mathrm{r} = 1,\ldots,\mathrm{z} \neq \mathrm{p} \\ 0 \leq k_i^p \leq D; \forall i \end{cases} \tag{3.15}$$

Čia p-tąją klasę atstovaujanti situacija pasirinkta pati pirmoji.

Sutvarkius 3.15 formulę gaunama 3.16 formulė.

$$\Phi^p = \sum_{i=1}^{N} k_i^p \cdot a_{1,i}^p \to \max$$

$$\begin{cases} \sum_{i=1}^{N} k_i^p (a_{1,i}^p \cdot \gamma - a_{lp,i}^p) \leq 0; \qquad \mathrm{lp} = 1,\ldots,\mathrm{Lp} \\ \sum_{i=1}^{N} k_i^p (a_{lr,i}^r - \kappa \cdot a_{1,i}^p) \leq 0; \qquad \mathrm{lr} = 1,\ldots,\mathrm{Lr}; \qquad \mathrm{r} = 1,\ldots,\mathrm{z} \neq \mathrm{p} \\ 0 \leq k_i^p \leq D; \forall i \end{cases} \tag{3.16}$$

Matosi, kad maksimizuojama panašumo funkcija – tai paprasta tiesinė funkcija, pagrįsta koreliacijos veiksmu, o pats uždavinys – gerai matematikoje išnagrinėtas tiesinio programavimo uždavinys, kuris yra paprastesnis nei gabalais tiesinio programavimo ir yra efektyvus bei tenkina resursų taupymo reikalavimą [9].

3.3.2. Atpažinimo algoritmo formulavimas

Atpažinimo algoritmas turi naudoti kiekvienos klasės savybių reikšmingumo koeficientus bei maksimizuojamos funkcijos reikšmę, apskaičiuotus su suformuluotu tiesinio programavimo uždaviniu (3.16). Nauja situacija priklausys tai klasei, į kurią bus labiausiai panaši. Todėl įvedama panašumo mato atpažinimo funkcija $\Phi(S, K^p)$ (p=1,...Z). Čia S – nauja, dar neatpažinta situacija. Atpažįstant naujas situacijas, norėtųsi, kad atpažinimo funkcija būtų tiesinė, kaip ir savybių reikšmingumo koeficientų radimo uždavinyje. Su kiekvienos klasės koeficientais apskaičiuotas funkcijos reikšmes reikia sunorminti su konstantoms, apskaičiuotoms pagal (3.12), ir iš visų sunormintų reikšmių išskirti maksimalią:

$$\max(c^p \cdot \Phi(S^x, K^p)), \quad \forall \mathrm{p} \tag{3.17}$$

Nauja situacija priklausys tai klasei, su kurios koeficientais sunorminta funkcija įgavo didžiausią reikšmę.

3.4. Kreditavimo proceso įvertinimo organizavimas

Kredito rizikai vertinti naudojami įvairūs ekspertiniai, statistiniai ir dirbtinio intelekto metodai bei jų modifikacijos [10, 11]. Statistiniais ir dirbtinio intelekto metodais vertinama patyriminė informacija, atsižvelgiant į sukauptus duomenis. Vertinant kredito riziką, didelę reikšmę turi vertintojo patirtis, nes dažniausiai ir vadovaujamasi savo patyriminę nuojauta. Problemų kyla tuomet, kai ekspertinio vertinimo, pagal kurio sukauptą patirtį nustatomos kreditavimo taisyklės, rezultatai akivaizdžiai nesutampa su statistiniais ir/ar dirbtinio intelekto metodais pagrįstų vertinimo modelių rezultatais. Todėl galima daryti prielaidą, kad dažnai kreditavimo taisyklės nėra adekvačios vertinimo praktikai, tačiau tik su sąlyga, kad turimame kreditų portfelyje yra santykinai mažai blogų kontraktų.

Ši problema gali būti sprendžiama pasitelkiant vieną iš dirbtinių neuroninių tinklų metodų – savitvarkius tinklus (SOM) [10, 11]. Palyginti su statistiniais klasterizavimo metodais, SOM išsiskiria efektyviu didelio duomenų kiekio apdorojimu, organizavimu ir vizualizacijos savybėmis. Tačiau SOM algoritmas klasterizavimą atlieka visam duomenų rinkiniui, o ne kiekvienai klasei atskirai. Todėl apskaičiuotus klasterius sudaro geri ir blogi kontraktai, tik gerų ir blogų kontraktų santykis atskiruose klasteriuose skiriasi. Toks sprendimo būdas pastaruoju metu jau netinka, nes jis nagrinėja tik panašumus tarp klasių, neatsižvelgdamas į skirtumus. Laikas naudoti intelektualesnius ir jautresnius metodus, nes situacijos iš skirtingų klasių gali būti labai panašios, bet skirtis tik vienu parametru. Lyginant tik jų panašumą, atrodys, kad jos labai panašios. Tačiau galbūt tas vienintelis parametras ir yra esminis, kad atskirtumėme klases. Todėl būtinai reikia įvertinti ir skirtumus tarp situacijų iš skirtingų klasių.

Literatūroje [1] žinoma automatinių kreditavimo sistemų, kurių veikimo principas paremtas instrukcijomis (taisyklėmis). Tokios sistemos dažniausiai pateikia grafinę vartotojo sąsają, kurią naudodami ekspertai paprastai sukuria atpažinimo modelį, susidedantį iš tam tikro kiekio taisyklių, indikuojančių, kaip sistemai elgtis vienu ar kitu atveju. Tačiau kartais ekspertams sudėtinga numatyti visus kreditavimo atvejus, be to, dažnai situacijos būna per daug miglotos ir instrukcijų kūrimas beveik neįmanomas.

Kredito kompanijos kontraktų vertinimui gali pateikti įvairių tipų parametrų, iš kurių susideda tie kontraktai. Keletas galimų parametrų pavyzdžių pateikti 3.7 lentelėje.

3.7 lentelė. Kontraktų parametrų pavyzdžiai

Parametro Nr.	Parametro pavadinimas
C_1	Kontrakto suma
C_2	Pradinis įnašas
C_3	Likutinė vertė
C_4	Kontrakto tipas
C_5	Kontrakto trukmė
C_6	Palūkanų procentas
C_7	Objekto tipas
C_8	Kliento tipas
C_9	Kliento valstybė
C_{10}	Atmestų kreditų skaičius
C_{11}	Įsiskolinimas
C_{12}	Mėnesinė alga

Čia C_1 – pradinė kredituojamo objekto vertė; C_2 – pradinio įnašo vertė; C_3 – likutinė objekto vertė po sutarties pasibaigimo; C_4 – kontrakto tipas (išperkamoji nuoma, verslo nuoma); C_5 – laiko tarpas, kuriam pasirašomas kontraktas; C_6 – palūkanų dydis procentais; C_7 – objekto tipas (lengvasis automobilis); C_8 – kliento tipas (privatus, kompanija); C_9 – valstybė, kurioje klientas gyvena; C_{10} – kreditų prašymų skaičius, kurie buvo atmesti; C_{11} – dabartinis kliento įsiskolinimas; C_{12} – kliento uždarbis per mėnesį.

3.5. Tikslo ir uždavinių detalizavimas

Atlikus literatūros analizę ir teorinius apibendrinimus, galima suformuluoti tokį darbo tikslą.

Tikslas – sukurti miglotosios situacijos įvertinimo modulį, kurį būtų lengva integruoti bet kurioje su kreditavimo procesais susijusioje kompanijoje, ir kuris iš kontraktų apdorojimo istorijos ir ypač iš pasibaigusių kontraktų apskaičiuotų parametrų koeficientus, įvertindamas ir jų tarpusavio sąryšius, ir panašumus bei skirtumus tarp sėkmingų ir nesėkmingų kontraktų.

Tikslui pasiekti reikia išspręsti tokius uždavinius:

1) atlikti visų situacijų norminimą tam, kad situacijų savybių įverčiai būtų tarpusavyje išmatuojami ir palyginami;
2) atlikti kiekvienos klasės situacijų grupavimą pagal jų tarpusavio panašumą tam, kad sumažinti situacijų kiekį klasėje;
3) sudaryti ir realizuoti taikymo sričiai (patyriminės istorijos kontraktams) tinkamą klasterizavimą;

4) realizuoti tiesinio programavimo uždavinio formulavimo metodą, atitinkantį taikymo sritį ir įvertinantį panašumus tarp tos pačios klasės situacijų bei skirtumus tarp skirtingų klasių situacijų;
5) realizuoti atpažinimo algoritmą, naudojant situacijų savybių reikšmingumo koeficientus;
6) sudaryti miglotai aprašytos situacijos įvertinimo modulio funkcinę organizaciją;
7) ištirti sudarytą modulį su:
   a) atsitiktinai sugeneruotais duomenimis,
   b) realiais istorinių kontraktų duomenimis.

4. MIGLOTOSIOS SITUACIJOS ĮVERTINIMO MODULIO REALIZACIJA

4.1. Modulio funkcinė organizacija

4.1.1. Realių kontraktų parametrų detalizavimas

Paprastai kompanijos turi sukaupusios didelius kiekius patyriminių kontraktų (situacijų). Esant tokioms sąlygoms ekspertams gali būti sudėtinga įvertinti visas situacijas ir jas suklasifikuoti, kad būtų įmanoma išspręsti savybių reikšmingumo koeficientų radimo uždavinį (plačiau 3.3.1 skyriuje). Todėl kyla poreikis prie kiekvienos dar neklasifikuotos situacijos prijungti išreikštinius parametrus, kurie dažniausiai sužinomi jau kontraktui pasibaigus, pavyzdžiui, gautas pelnas, mokėjimų atidėliojimų skaičius ir panašiai. Tuo būdu atsitinka taip, kad situacijos aprašas susidaro iš dviejų tipų parametrų: pirminių ir išreikštinių. Tarkim, egzistuoja L patyriminių situacijų, kurios atrodo taip:

$$\begin{cases} S_1 = \{\overbrace{a_{1,1} \quad a_{1,2} \quad \dots \quad a_{1,m-1} \quad a_{1,m}}^{\textit{Pirminiai parametrai}} \qquad \overbrace{b_{1,1} \quad b_{1,2} \quad \dots \quad b_{1,n-1} \quad b_{1,n}}^{\textit{Išreikštiniai parametrai}}\} \\ \dots \\ S_L = \{a_{L,1} \quad a_{L,2} \quad \dots \quad a_{L,m-1} \quad a_{L,m} \qquad b_{L,1} \quad b_{L,2} \quad \dots \quad b_{L,n-1} \quad b_{L,n}\} \end{cases} . \tag{4.1}$$

Čia m – pirminių parametrų skaičius; n – išreikštinių parametrų skaičius.

Pirminiai parametrai – tai informacija apie klientą (pavyzdžiui, mėnesinis atlyginimas, kapitalas ir daug kitų). Pagal šiuos parametrus reikia apskaičiuoti jų reikšmingumo koeficientus, naudojamus naujų situacijų atpažinime. Išreikštiniai parametrai reikalingi situacijoms klasifikuoti. Verslo ekspertai taip pat nurodo klasių kiekį ir bent po vieną tipinį atstovą pagal išreikštinius parametrus kiekvienai klasei:

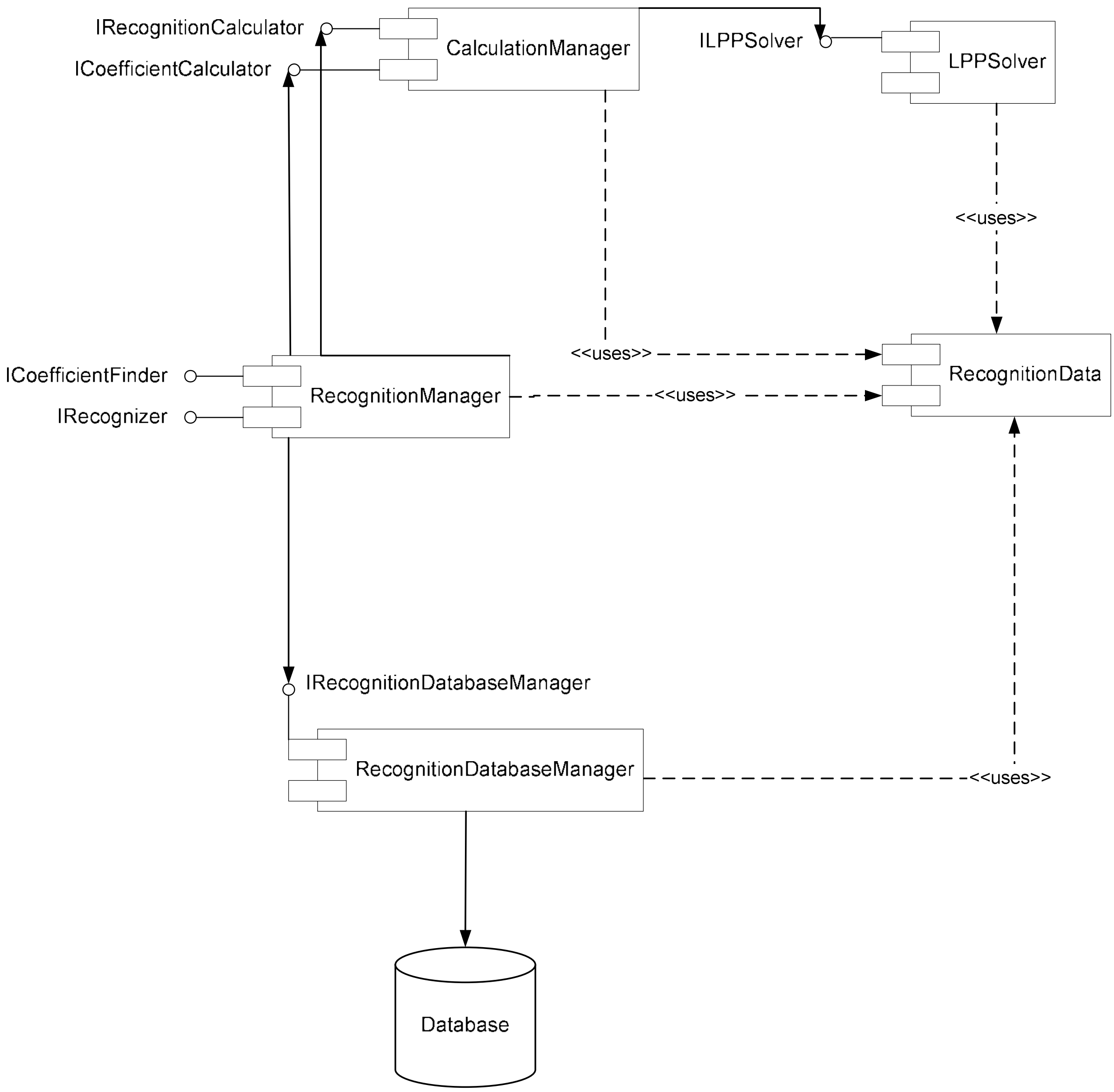
$$\overbrace{T^p_{dp} = \{b^p_{dp,1} \quad b^p_{dp,2} \quad \dots \quad b^p_{dp,n-1} \quad b^p_{dp,n}\}}^{\textit{Tipiniai atstovai}}; \mathrm{dp} = 1,\dots,\mathrm{Dp};\ \mathrm{p} = 1,\dots,\mathrm{Z} \tag{4.2}$$

Čia Dp – p-tosios klasės tipinių atstovų pagal išreikštinius parametrus skaičius; Z – klasių skaičius. Žinant tipinius atstovus, reikia kiekvienai klasei apskaičiuoti išreikštinių parametrų reikšmingumo koeficientus. Paskui visa patyriminių istorijų aibė suklasifikuojama naudojant išskaičiuotuosius koeficientus. Ir po to išskaičiuojami kliento parametrų reikšmingumo koeficientai kiekvienai klasei. Pastarieji jau gali būti naudojami naujų situacijų (kontraktų) atpažinime.

4.1.2. Komponentų diagrama

Remiantis išdėstyta medžiaga savybių reikšmingumo koeficientų radimo uždavinio (3.3.1) bei atpažinimo algoritmo formulavimo (3.3.2) skyriuose, miglotai apibrėžtos situacijos įvertinimo modulis buvo sudarytas iš šių komponentų (4.1 pav.):

1) „RecognitionManager“ (liet. *Atpažinimo valdymas*),
2) „CalculationManager“ (liet. *Skaičiavimų valdymas*),
3) „LPPSolver“ (liet. *Tiesinio programavimo uždavinio (TPU) sprendimas*),
4) „RecognitionDatabaseManager“ (liet. *Atpažinimo duomenų bazės valdymas*),
5) „RecognitionData“ (liet. *Atpažinimo duomenų struktūros*),
6) „Database“ (liet. *Duomenų bazė*).



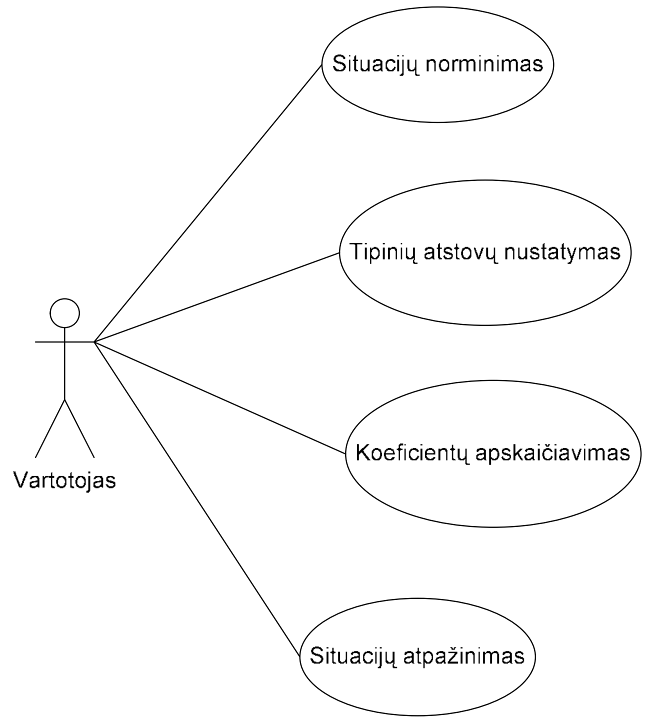
4.1 pav. Komponentų diagrama

„RecognitionManager“ komponentas valdo visą modulį ir realizuoja dvi sąsajas: „ICoefficientFinder“ ir „IRecognizer“, kurios leidžia iš grafinės vartotojo sąsajos atitinkamai atlikti koeficientų apskaičiavimą ir/ar atpažinimą. Šis komponentas – tai išorinis priėjimas prie modulio. Visi kiti komponentai – vidiniai. „RecognitionManager“ naudoja „CalculationManager“, kuris realizuoja taip pat dvi sąsajas: „ICoefficientCalculator“, skirtą pirminių parametrų reikšmingumo koeficientų radimo uždaviniui spręsti; „IRecognitionCalculator“ – atpažinimo uždaviniui spręsti. „CalculationManager“ per „ILPPSolver“ sąsają naudoja „LPPSolver“ komponentą, kuris formuluoja tiesinio programavimo uždavinį ir naudoja tam tikrą biblioteką, kuri tą uždavinį išsprendžia. „RecognitionDatabaseManager“ skirtas prieiti prie duomenų bazės. Joks kitas komponentas neturi tiesioginio priėjimo prie duomenų saugyklos. „RecognitionData“ reikalingas aprašyti toms duomenų struktūroms, kurias naudoja modulio komponentai. „Database“ – duomenų bazė, kurioje saugomi sistemos funkcionavimui būtini duomenys.

4.1.3. Panaudos atvejai

Kreditavimo procesų analizė rodo, kad miglotai aprašytos situacijos įvertinimo modulis turi keturis pagrindinius panaudos atvejus (4.2 pav.):

1) situacijų norminimas;
2) tipinių atstovų nustatymas;
3) koeficientų apskaičiavimas;
4) situacijų atpažinimas.

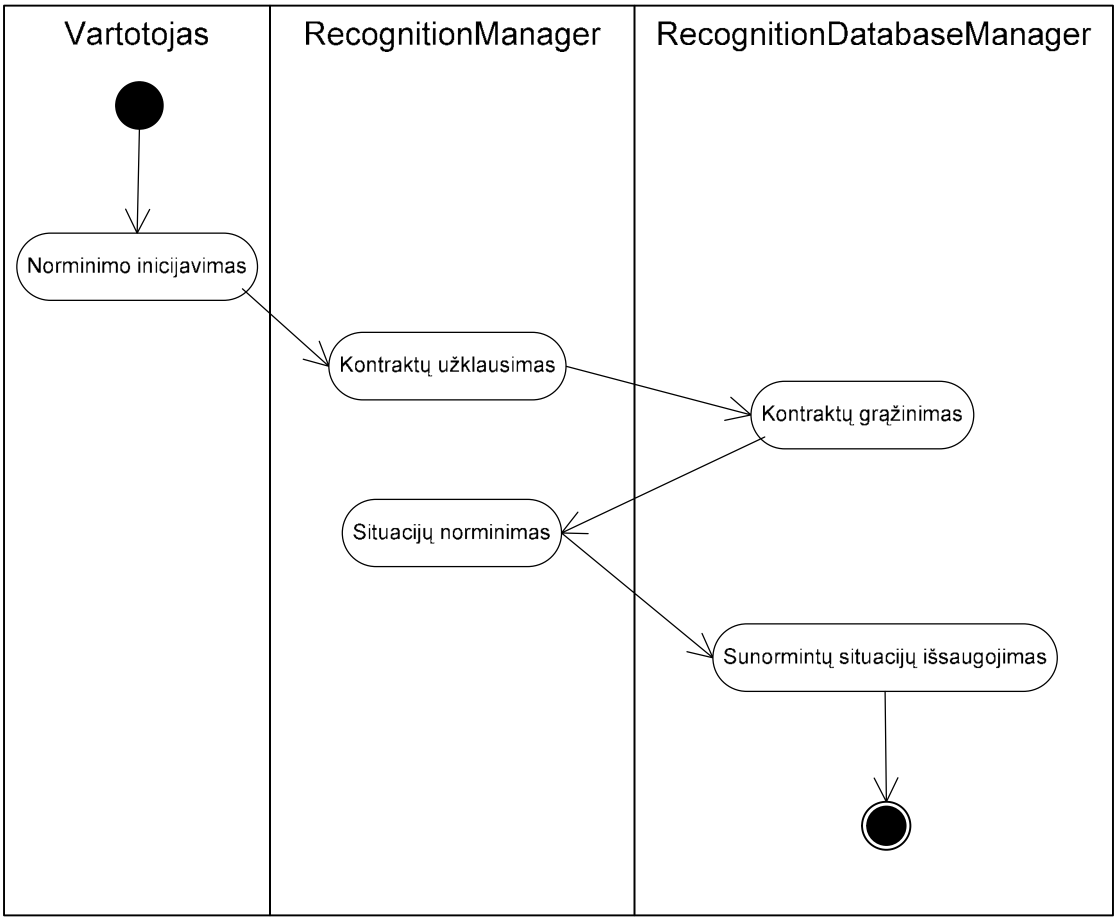


4.2 pav. Panaudos atvejai

Naudodamasis tuo moduliu, vartotojas gali atlikti patyriminių situacijų norminimą, nustatyti kiekvienos klasės išreikštinių parametrų tipinius atstovus, apskaičiuoti visų klasių pirminių parametrų reikšmingumo koeficientus bei atpažinti naujas situacijas. Modulio panaudos atvejai detaliau atvaizduojami veiklų diagramomis.

4.1.4. Situacijų norminimo veiklų diagrama

Situacijų norminimo veiklų diagrama pavaizduota 4.3 paveiksle.



4.3 pav. Situacijų norminimo veiklų diagrama

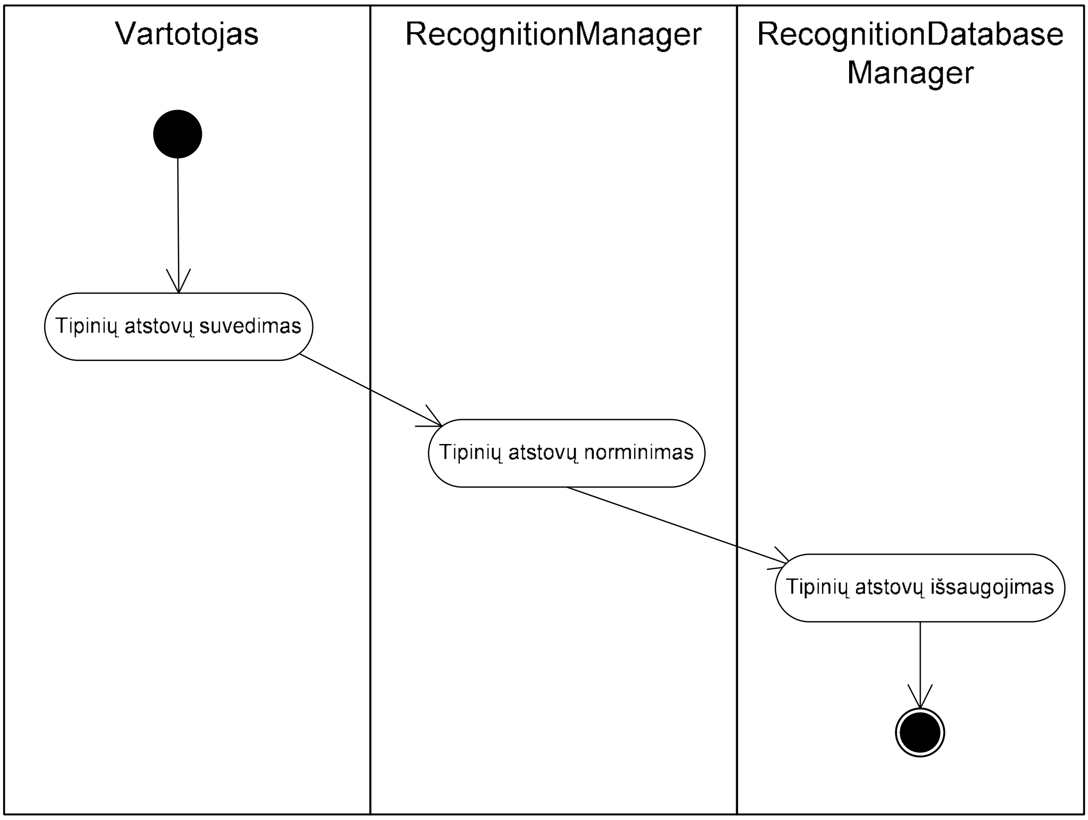
Veiklų aprašymas:

- „Norminimo inicijavimas“ – vartotojas parenka norminimo intervalą, nustato, kuriuos kontraktus norminti, ir inicijuoja patį norminimą;
- „Kontraktų užklausimas“ – „RecognitionManager“ komponentas pareikalauja reikiamų kontraktų (situacijų);
- „Kontraktų gražinimas“ – „RecognitionDatabaseManager“ komponentas gražina duomenis pagal pateiktą užklausą;
- „Situacijų norminimas“ – „RecognitionManager“ atlieka situacijų parametrų norminimą (3.1 skyrius);

* „Sunormintų situacijų išsaugojimas“ – „RecognitionDatabaseManager“ išsaugo sunormintas situacijas duomenų bazėje.

4.1.5. Tipinių atstovų nustatymo veiklų diagrama

Tipinių atstovų nustatymo veiklų diagrama pavaizduota 4.4 paveiksle.



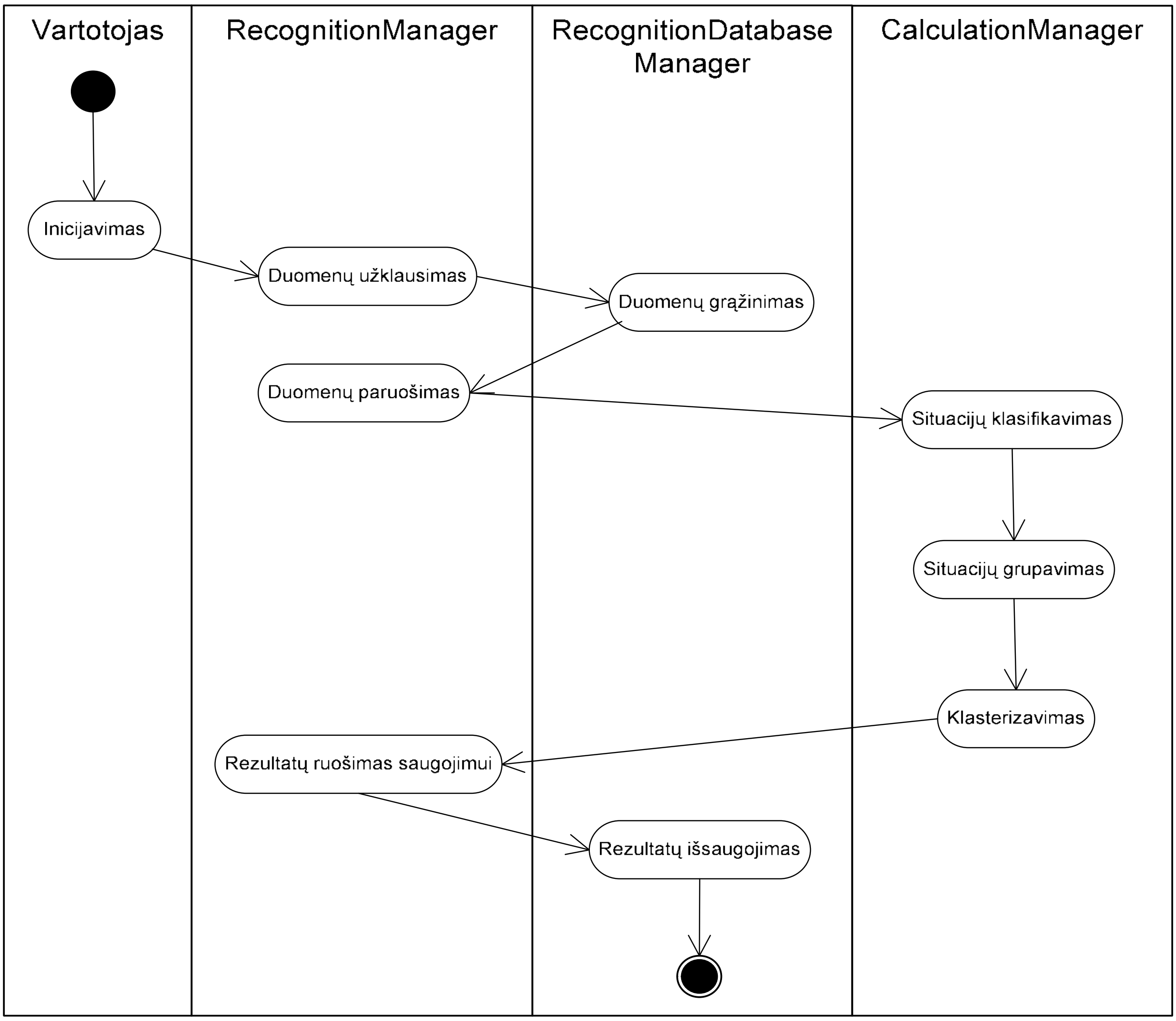
4.4 pav. Tipinių atstovų nustatymo veiklų diagrama

Veiklų aprašymas:

* „Tipinių atstovų suvedimas“ – vartotojas suveda kiekvienos klasės išreikštinių parametrų tipinius atstovus (4.2);
* „Tipinių atstovų norminimas“ – tipinių atstovų parametrų norminimas, kurį atlieka komponentas „RecognitionManager“ (3.1 skyrius);
* „Tipinių atstovų išsaugojimas“ – „RecognitionDatabaseManager“ išsaugo tipinius atstovus duomenų bazėje.

4.1.6. Koeficientų apskaičiavimo veiklų diagrama

Koeficientų apskaičiavimo veiklų diagrama pavaizduota 4.5 paveiksle.

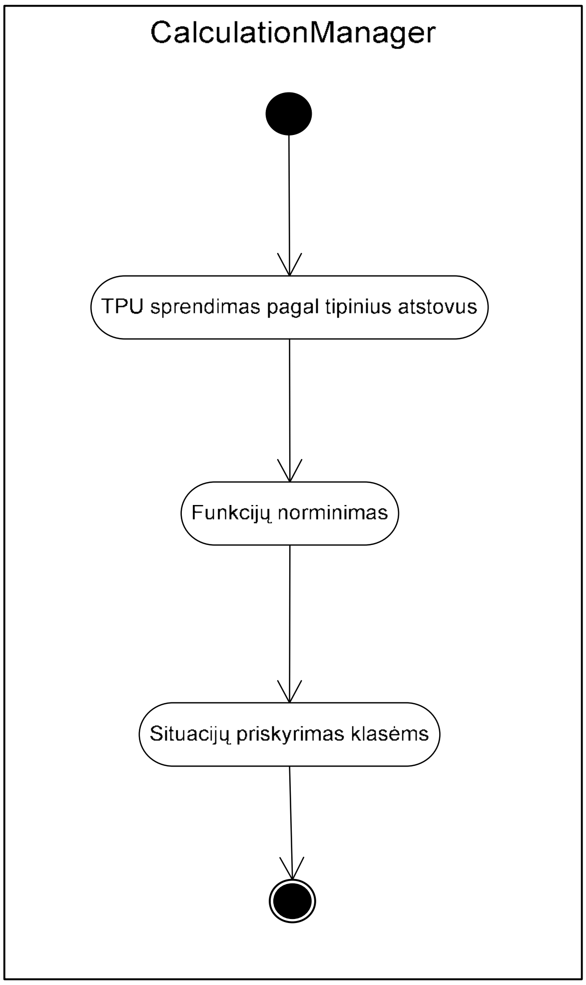


4.5 pav. Koeficientų apskaičiavimo veiklų diagrama

Veiklų aprašymas:

- „Inicijavimas“ – vartotojas inicijuoja pirminių parametrų reikšmingumo koeficientų skaičiavimo uždavinį;
- „Duomenų užklausimas“ – „RecognitionManager“ pareikalauja reikiamų situacijų su pirminiais ir išreikštiniais parametrais (4.1), patyriminių klasių bei kiekvienos klasės tipinių atstovų (4.2), sudarytų tik iš išreikštinių parametrų;
- „Duomenų grąžinimas“ – „RecognitionDatabaseManager“ išrenka ir gražina duomenis iš duomenų bazės pagal pateiktą užklausą. Grąžinamos situacijos bei tipiniai klasių atstovai būna jau sunorminti, nes šis procesas vykdomas tik po situacijų norminimo (plačiau 4.1.4 skyriuje) ir po tipinių atstovų suvedimo (plačiau 4.1.5 skyriuje);

- „Duomenų paruošimas“ – „RecognitionManager“ gautas situacijas bei klasių tipinius atstovus centruoja, paruošia bei pateikia „CalculationManager“ komponentui reikiama forma;
- „Situacijų klasifikavimas“ – atliekamas komponente „CalculationManager“ kol dar situacijos nepriklauso jokiai klasei (situacijos neklasifikuotos). Šį procesą galima pavaizduoti tokia schema (4.6 pav.):



4.6 pav. Situacijų klasifikavimo veiklų diagrama

Situacijų klasifikavimo veiklų aprašymas:

✓ „TPU sprendimas pagal tipinius atstovus“. Patyriminių klasių tipiniai atstovai – tai situacijos, sudarytos tik iš išreikštinių parametrų (4.2). Todėl pagal tuos tipinius atstovus suformuluojami tiesinio programavimo uždaviniai (plačiau apie TPU formulavimą 3.3.1 skyriuje) ir apskaičiuojami išreikštinių parametrų reikšmingumo koeficientai (išreikštiniai koeficientai) bei maksimizuojamos funkcijos reikšmės kiekvienai klasei:

$$\overbrace{K^p = \{k_1^p \quad k_2^p \quad \ldots \quad k_{n-1}^p \quad k_n^p\}}^{\textit{Išreikštiniai koeficientai}} \qquad \overbrace{\Phi_{\max}^p}^{\textit{Maksimizuojama funkcija}} \qquad ;\, \mathrm{p} = 1,\ldots,Z\,. \quad (4.3)$$

Čia Z – klasių skaičius; n – išreikštinių parametrų skaičius.

✓„Funkcijų norminimas" – visų klasių maksimizuojamos funkcijos sunorminamos (3.12) tam, kad atpažįstant situaciją pagal išreikštinius parametrus (priskiriant situaciją kuriai nors klasei), rezultatai būtų palyginami.

✓„Situacijų priskyrimas klasėms" – pagal kiekvienos dar neklasifikuotos situacijos išreikštininius parametrus bei gautus koeficientus formuluojamas atpažinimo uždavinys (3.3.2 skyrius). Tokių atpažinimo uždavinių skaičius lygus situacijų kiekiui. Klasifikavime naudojamas tas pats atpažinimo uždavinys, kuris naudojamas ir situacijų atpažinime pagal pirminius parametrus. Situacija priskiriama tai klasei, į kurią yra panašiausia.

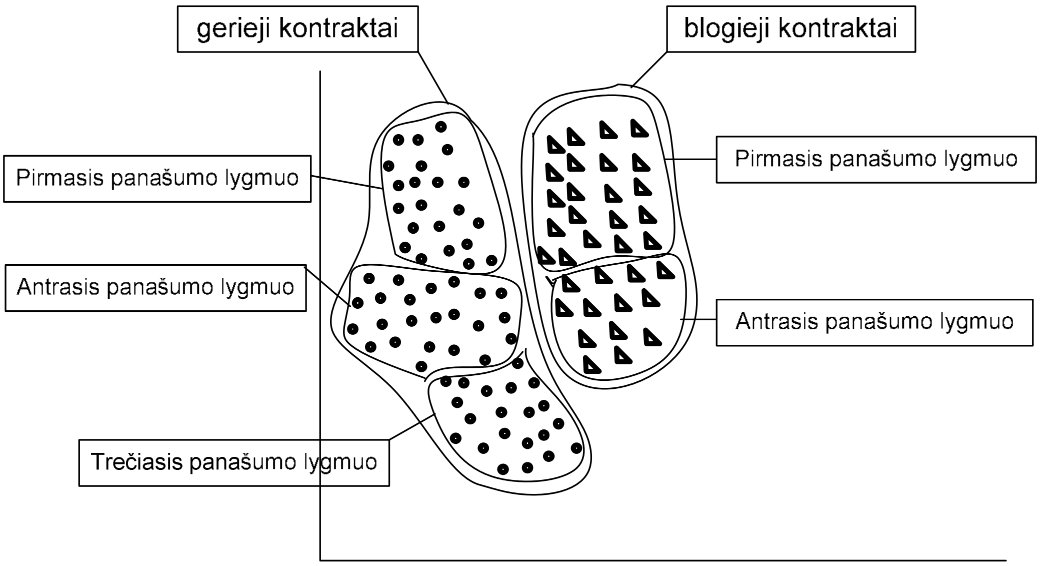
Po klasifikavimo situacijos, aprašytos 4.1 formulėje, jau priklauso kuriai nors klasei:

$$\left\{ S_{lp}^{p} = \{ \overbrace{a_{lp,1}^{p} \quad a_{lp,2}^{p} \quad \ldots \quad a_{lp,m-1}^{p} \quad a_{lp,m}^{p}}^{Pirminiai\ parametrai} \qquad \overbrace{b_{lp,1}^{p} \quad b_{lp,2}^{p} \quad \ldots \quad b_{lp,n-1}^{p} \quad b_{lp,n}^{p}}^{Išreikštiniai\ parametrai} \} \right. , \quad (4.4)$$

p = 1,...,Z; lp = 1,...,Lp. Čia p – klasių skaičius; Lp – p-tosios klasės situacijų skaičius. Visų situacijų skaičius L lygus visų klasių situacijų skaičių sumai:

$$L = \sum_{p=1}^{Z} Lp \tag{4.5}$$

- „Situacijų grupavimas" – tai situacijų kiekvienoje klasėje suskirstymas į panašumo lygmenis; tokio suskirstymo idėja iliustruojama 4.7 pav.



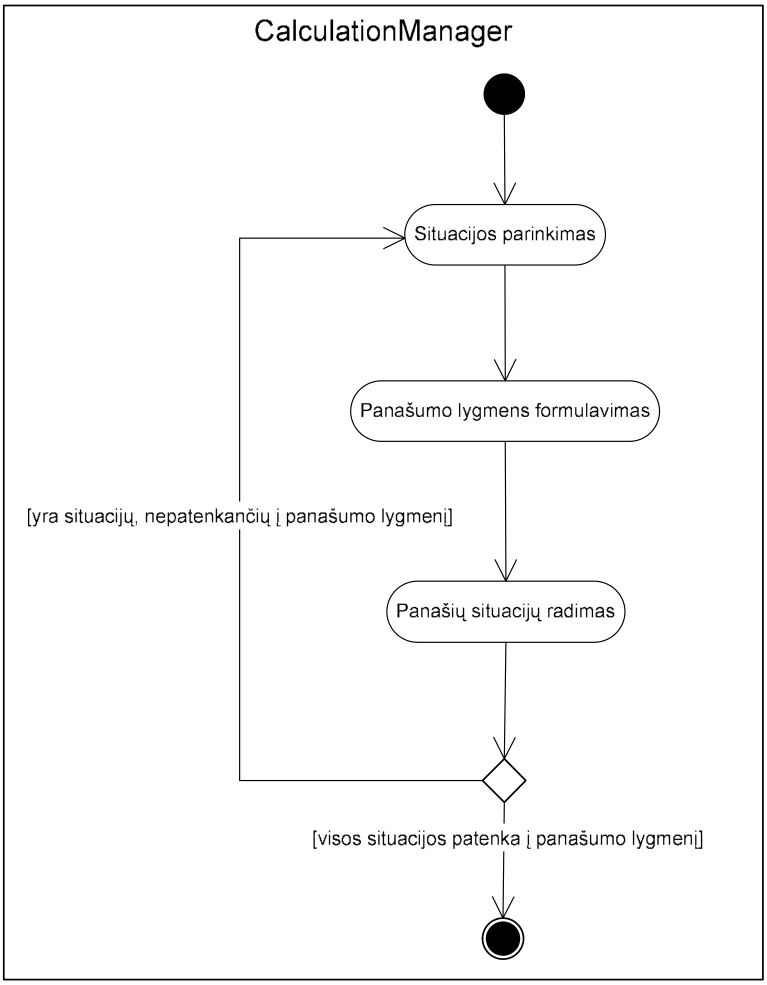
4.7 pav. Situacijų panašumo lygmenys

Paprastai patyriminės istorijos kontraktai būna išsibarstę, tai yra, nebūtinai visada pirmiau eina geri kontraktai, paskui blogi. Gerų ir blogų kontraktų atskyrimui naudojamas klasifikavimas. Tačiau netgi toje pačioje klasėje situacijos gali būti gana nepanašios. Kadangi jos yra išsibarsčiusios, tai tiesinio programavimo uždavinių, suformuluotų pagal iš eilės imamų situacijų pirminius parametrus, apibrėžimo sritis gali turėti tik vieną tašką – koordinačių pradžią, todėl visų koeficientų reikšmės gautis 0. Tarkim taip galėtų atsitikti, jeigu pirminių parametrų reikšmingumo koeficientų radimo uždavinyje iš gerų kontraktų klasės būtų naudojamos situacijos iš pirmo ir trečio panašumo lygmens, nes suformuluotas tiesinio programavimo uždavinys gali netenkinti sąlygos, kad situacijos iš vienos klasės būtų γ-panašios (plačiau apie savybių reikšmingumo koeficientų radimo uždavinį 3.3.1 skyriuje).

Kad išvengti tokių netikėtumų, kai uždavinys duoda nulinius rezultatus, daromas situacijų grupavimas į panašumo lygmenis. Grupavime naudojami tik pirminiai klasifikuotų situacijų parametrai:

$$\left\{ S_{lp}^{p} = \{\overbrace{a_{lp,1}^{p} \quad a_{lp,2}^{p} \quad \ldots \quad a_{lp,m-1}^{p} \quad a_{lp,m}^{p}}^{Pirminiai\ parametrai}\};\ \forall \mathrm{p}; \forall \mathrm{lp}. \right. \tag{4.6}$$

Situacijų grupavimas vaizduojamas tokia schema (4.8 pav.):



4.8 pav. Situacijų grupavimo veiklų diagrama

Situacijų grupavimo veiklų aprašymas:

- ✓„Situacijos parinkimas" – parenkama atstovaujanti panašumo lygmenį situacija (dažniausiai pati pirmoji jokiam panašumo lygmeniui dar nepriklausanti klasės situacija).
- ✓„Panašumo lygmens formulavimas" – sukuriamas klasės panašumo lygmuo, kurį atstovauja parinkta situacija.
- ✓„Panašių situacijų radimas" daromas paimant atstovaujančios panašumo lygmenį situacijos pirminių parametrų įverčius ir lyginant juos su kitų tos pačios klasės situacijų, kurios dar nepriklauso jokiam panašumo lygmeniui, parametrų įverčiais. Patogu tokiu lyginimo įverčiu laikyti koreliacijos koeficientą:

$$\pi_{hg} = \frac{\sum_{i=1}^{m}(a_{h,i}^{p} \cdot a_{g,i}^{p})}{\sqrt{\sum_{i=1}^{m}(a_{h,i}^{p})^2} \cdot \sqrt{\sum_{i=1}^{m}(a_{g,i}^{p})^2}}. \tag{4.7}$$

Čia π_{hg} – panašumas; $a_{h,i}^{p}$ – p-tosios klasės h-tosios situacijos i-tojo parametro įvertis; m – pirminių parametrų kiekis. Situacijos patenka į vieną panašumo lygmenį, kai panašumas π_{hg} didesnis už užsibrėžtą ribą (pavyzdžiui, riba gali būti dydis γ).

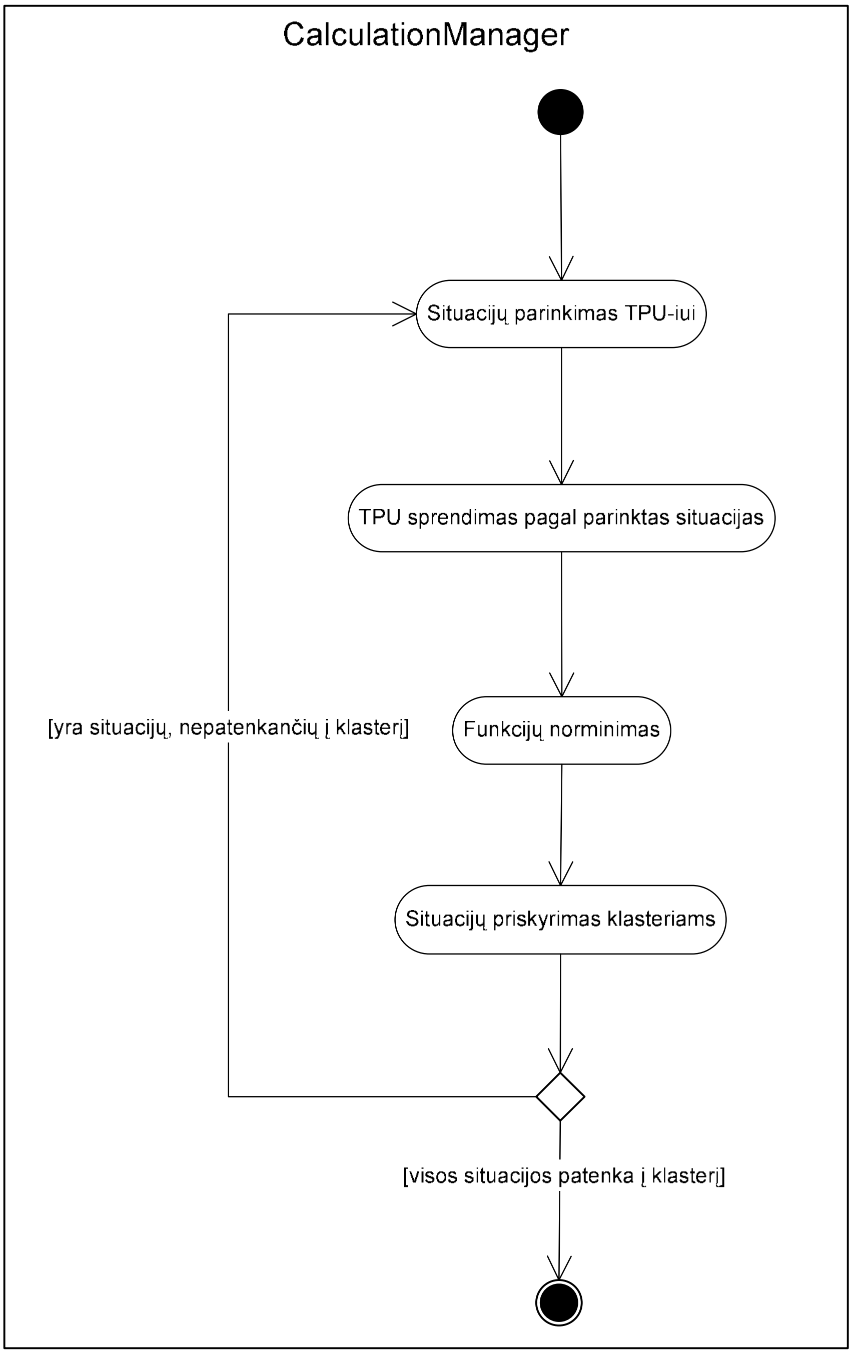
Situacijų grupavimas – iteracinis procesas, vykdomas tol, kol nebelieka klasėje situacijų, nepatenkančių į kurį nors panašumo lygmenį. Panašumo lygmeniui gali priklausyti ir viena situacija, jei ji nepanaši nei į vieną bet kurią kitą situaciją, nepriklausančią kitam panašumo lygmeniui. Viename lygmenyje situacijos yra γ-panašios, bet pats panašumas tarp lygmenų nėra matuojamas. Tai yra, dvi situacijos iš skirtingų panašumo lygmenų taip pat gali būti γ-panašios, o jų patekimas į skirtingus lygmenis priklauso nuo pasirinktos panašumo lygmenį atstovaujančios situacijos, su kuria viena iš jų yra, o kita nėra γ-panašios. Panašumo lygmenų skirtingose klasėse skaičius gali skirtis, nes jis priklauso nuo tos klasės situacijų panašumų tarpusavyje. Pavyzdžiui, jei situacijos tarpusavyje labai panašios, tai gali užtekti ir vieno panašumo lygmens, o jei mažiau panašios, tai daugiau panašumo lygmenų.

Po grupavimo situacijos atrodo taip (situacijos vaizduojamos tik su pirminiais parametrais, nes tik pagal juos vykdomas grupavimas):

$$\left\{ S_{lp}^{p,gp} = \{\overbrace{a_{lp,1}^{p,gp} \quad a_{lp,2}^{p,gp} \quad \ldots \quad a_{lp,m-1}^{p,gp} \quad a_{lp,m}^{p,gp}}^{Pirminiai\ parametrai}\};\ \forall \mathrm{p}; \forall \mathrm{lp};\ \mathrm{gp} = 1,\ldots,\mathrm{Gp}. \right. \tag{4.8}$$

Čia $S_{lp}^{p,gp}$ – p-tosios klasės gp-tojo panašumo lygmens lp-toji situacija; Gp – p-tosios klasės panašumo lygmenų skaičius.

- „Klasterizavimas“ – atliekamas komponente „CalculationManager“ bei, kaip ir situacijų grupavimas, naudoja tik pirminius patyriminių situacijų parametrus (4.6). Šį procesą galima pavaizduoti tokia veiksmų schema (4.9 pav.):



4.9 pav. Klasterizavimo veiklų diagrama

Klasterizavimo veiklų aprašymas:

✓ „Situacijų parinkimas TPU-iui“ – tai procesas, kurio metu iš sugrupuotų klasių parenkamos situacijos (plačiau apie situacijų parinkimą 4.2 skyriuje), kurios bus naudojamos formuluoti pirminių parametrų reikšmingumo koeficientų (pirminių koeficientų) radimo uždavinius klasėms (3.16).

- ✓ „TPU sprendimas pagal parinktas situacijas“ – pagal parinktas situacijas suformuluojami tiesinio programavimo uždaviniai ir apskaičiuojami kiekvienos p-tosios klasės wp-tasis klasteris, sudarytas iš pirminių parametrų reikšmingumo koeficientų (pirminių koeficientų) bei maksimizuojamų funkcijų reikšmių:

$$K_{wp}^{p} = \overbrace{\{k_{wp,1}^{p} \quad k_{wp,2}^{p} \quad \ldots \quad k_{wp,m-1}^{p} \quad k_{wp,m}^{p}\}}^{\textit{Pirminiai koeficientai}} \qquad \overbrace{\Phi_{wp\,\max}^{p}}^{\textit{Maksimizuojama funkcija}} \quad , \ \mathrm{p} = 1,\ldots,\mathrm{Z} \quad (4.9)$$

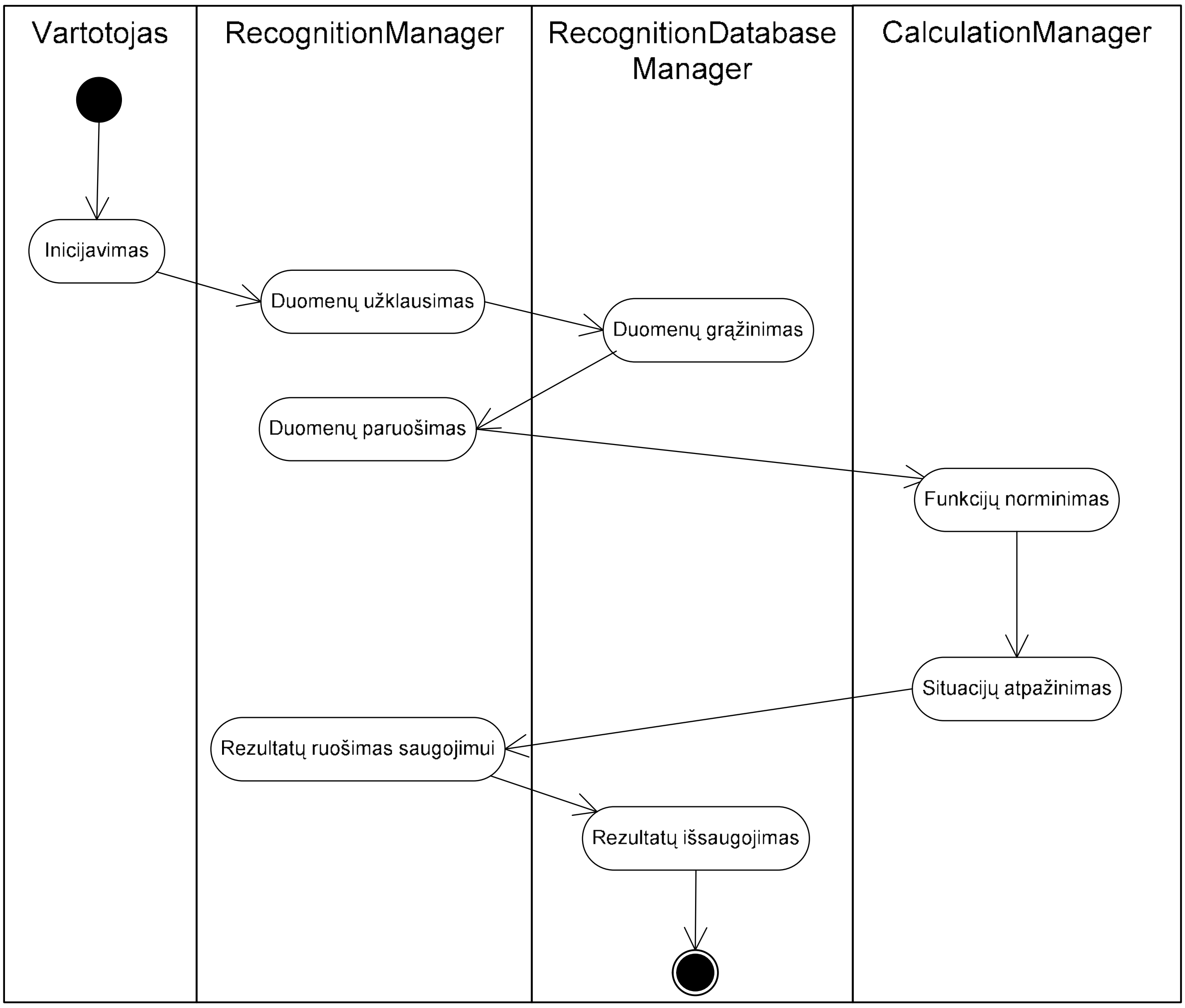
Čia K_{wp}^{p} – p-tosios klasės wp-tojo klasterio pirminių koeficientų reikšmių rinkinys; $\Phi_{wp\,\max}^{p}$ – p-tosios klasės wp-tojo klasterio maksimizuojamos funkcijos reikšmė.

- ✓ „Funkcijų norminimas“ – visų klasterių maksimizuojamos funkcijos norminamos (3.12) tam, kad atpažįstant situaciją pagal pirminius parametrus (tikrinant, ar pagal apskaičiuotus klasterius situacija panašiausia į savo klasę), rezultatai būtų palyginami.
- ✓ „Situacijų priskyrimas klasteriams“. Su kiekviena situacija bei apskaičiuotų klasterių pirminiais koeficientais formuluojamas atpažinimo uždavinys (3.3.2 skyrius). Kadangi situacijos jau klasifikuotos, tai šio proceso metu tikrinama, ar visos situacijos pagal apskaičiuotus klasterius tikrai panašiausios į savo klases. Jeigu panašiausios, tuomet jos laikomos atpažintos. Jeigu situacija panašesnė ne į tą klasę, kuriai priklauso, tai ji laikoma neatpažinta. Klasterizavimas – tai iteracinis procesas, vykdomas tol, kol yra bent viena neatpažinta situacija. Klasterių skaičius prieš klasterizuojant nėra apibrėžtas ir gali kisti priklausomai nuo tos pačios klasės situacijų panašumo tarpusavyje bei nepanašumo tarp situacijų iš skirtingų klasių. Be to, klasterių skaičius skirtingose klasėse gali būti taip pat skirtingas, nes kurios nors klasės visos situacijos gali būti jau atpažintos, o kitų klasių kai kurios situacijos dar neatpažintos.

- „Rezultatų ruošimas saugojimui“ – gauti rezultatai ruošiami saugojimui bei pateikiami „RecognitionDatabaseManager“ komponentui reikiama forma.
- „Rezultatų išsaugojimas“ – rezultatų išsaugojimas duomenų bazėje.

4.1.7. Situacijų atpažinimo veiklų diagrama

Situacijos atpažinimo veiklų diagrama pavaizduota 4.10 paveiksle.



4.10 pav. Situacijos atpažinimo veiklų diagrama

Veiklų aprašymas:

- „Inicijavimas“ – vartotojas inicijuoja naujų situacijų atpažinimo uždavinį.
- „Duomenų užklausimas“ – „RecognitionManager“ pareikalauja situacijų, sudarytų tik iš pirminių parametrų (išreikštinių parametrų net neturi, nes jos ne patyriminės, o naujos) (4.10), kurios dar bus atpažintos, patyriminių klasių bei kiekvienos klasės pirminių klasterių (4.9).

$$\begin{cases} S_1 = \{\overbrace{a_{1,1} \quad a_{1,2} \quad \dots \quad a_{1,m-1} \quad a_{1,m}}^{Pirminiai\ parametrai}\} \\ \dots \\ S_F = \{a_{F,1} \quad a_{F,2} \quad \dots \quad a_{F,m-1} \quad a_{F,m}\} \end{cases} . \quad (4.10)$$

Čia F – naujų situacijų skaičius.

- „Duomenų grąžinimas“ – „RecognitionDatabaseManager“ išrenka ir gražina duomenis iš duomenų bazės pagal pateiktą užklausą. Grąžinamos naujos situacijos būna jau sunormintos, nes šis procesas vykdomas tik po situacijų norminimo (plačiau 4.1.4 skyriuje).
- „Duomenų paruošimas“ – „RecognitionManager“ gautas situacijas centruoja. Po to jas su klasių klasteriais paruošia bei pateikia „CalculationManager“ komponentui reikiama forma;
- „Funkcijų norminimas“ – kiekvienos klasės visų klasterių maksimizuojamos funkcijos norminamos (3.12).
- „Situacijų atpažinimas“ – su kiekviena situacija bei pirminiais koeficientais formuluojamas atpažinimo uždavinys (3.3.2 skyrius).
- „Rezultatų ruošimas saugojimui“ – gauti rezultatai ruošiami saugojimui bei pateikiami „RecognitionDatabaseManager“ komponentui reikiama forma.
- „Rezultatų išsaugojimas“ – rezultatų išsaugojimas duomenų bazėje.

4.2. Klasterizavimo metodų realizacija

Paprastai kredito kompanijos būna sukaupusios didelį kiekį patyriminių situacijų (kontraktų). Kadangi visų patyriminių situacijų panaudojimas TPU spręsti reikalautų per daug kompiuterinių ir laiko resursų, todėl tos situacijos grupuojamos pagal panašumo lygmenis, ir TPU formuluotėse panaudojami tik tų grupių (lygmenų) tipiniai atstovai. Žodžiu, privalu mažinti situacijų, pagal kurias sudaromi tiesinio programavimo uždaviniai, kiekį. Pirmas situacijų mažinimo žingsnis – tai situacijų grupavimas (4.1.6 skyrius 4.8 pav.). Pats grupavimas situacijų skaičiaus nesumažina, o tik suskirsto jas į grupes su tuo pačiu panašumo lygmeniu ir paruošia jas klasterizavimui, kurio metu iš panašumo lygmenų atrenkamos tik tam tikros situacijos, paduodamos į TPU (4.1.6 skyrius 4.9 pav.). Kadangi situacijos iš vieno panašumo lygmens yra γ-panašios, tai galima pakankamai racionaliai iš tų lygmenų parinkti juos atstovaujančias situacijas, su kuriomis ir sprendžiamas TPU. Pagal gautus pirminių parametrų reikšmingumo koeficientus galima patikrinti, ar ir situacijos, kurios į TPU nebuvo įtrauktos, tikrai panašiausios į savo klases.

Šiame darbe tyrimuose realizuoti du pagrindiniai klasterizavimo metodai:

1) pirmas klasterizavimo metodas (4.2.1 skyrius);

2) antras klasterizavimo metodas (4.2.2 skyrius).

Abiejų klasterizavimo metodų apibendrintai vykdomų veiksmų seka, pavaizduota 4.9 paveiksle. Skiriasi tik situacijų atrinkimo būdas tiesinio programavimo uždaviniui spręsti.

4.2.1. Pirmasis klasterizavimo metodas

Pirmasis klasterizavimo metodas realizuotas tikintis, kad visos situacijos iš skirtingų panašumo lygmenų, yra γ-panašios. Todėl p-tajai klasei TPU formuluojamas (3.3.1 skyrius), paimant pirmą klasteriui dar nepriskirtą situaciją $S_{ntp}^{p,kp}$ iš pirmojo pasitaikiusio panašumo lygmens kp, kuriame dar yra klasteriams nepriskirtų situacijų. Pagal $S_{ntp}^{p,kp}$ sudaroma maksimizuojamoji funkcija. Po to imama po vieną situaciją iš kitų tos klasės panašumo lygmenų ir reikalaujama, kad jos su $S_{ntp}^{p,kp}$ būtų γ-panašios; taip pat imama po vieną situaciją iš kitų klasių panašumo lygmenų ir reikalaujama, kad jos su $S_{ntp}^{p,kp}$ būtų κ-nepanašios:

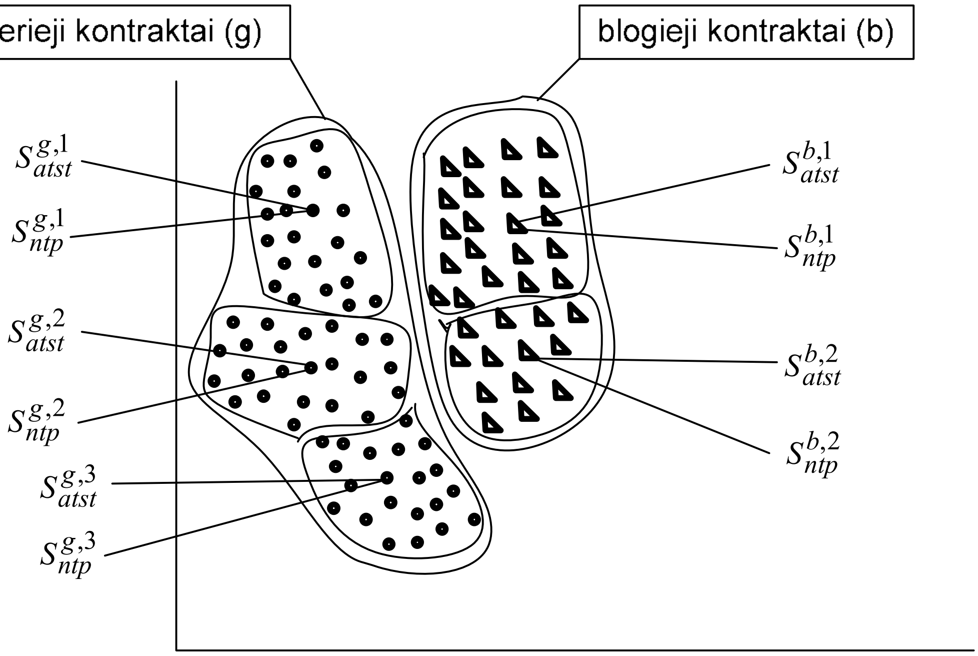
$$\Phi_{wp}^{p} = \sum_{i=1}^{m} k_{wp,i}^{p} \cdot a_{ntp,i}^{p,kp} \rightarrow \max$$

$$\begin{cases} \sum_{i=1}^{m} k_{wp,i}^{p} (a_{ntp,i}^{p,kp} \cdot \gamma - a_{atst,i}^{p,gp}) \leq 0; \qquad \text{gp} = 1,...,\text{Gp} \neq \text{kp} \\ \sum_{i=1}^{m} k_{wp,i}^{p} (a_{atst,i}^{r,gr} - \kappa \cdot a_{ntp,i}^{p,kp}) \leq 0; \qquad \text{gr} = 1,...,\text{Gr}; \qquad \text{r} = 1,...,\text{z} \neq \text{p} \\ 0 \leq k_{wp,i}^{p} \leq D; \qquad \forall i; \qquad \text{wp} = 1,...,\text{Wp} \end{cases} \tag{4.11}$$

TPU formuluojamas p-tajai klasei tik su sąlyga, kad toje klasėje dar yra bent vienas panašumo lygmuo, turintis bent vieną neatpažintą situaciją. Klasterizavimas baigiasi tada, kai nė viena klasė neturi panašumo lygmens, kuriame būtų bent viena neatpažinta situacija.

Sutrumpinimas *ntp* reiškia neatpažinta, o *atst* – atstovaujanti. Neatpažinta situacija $S_{ntp}^{p,kp}$ iš bet kurios p-tosios klasės bet kurio kp-tojo panašumo lygmens yra pati pirmoji tame lygmenyje pasitaikiusi situacija, kuri pagal apskaičiuotus klasterius nebuvo panašiausia į savo klasę. Jeigu klasteriai dar nebuvo skaičiuoti, tai $S_{ntp}^{p,kp}$ yra pati pirmoji situacija savo lygmenyje (panašumo lygmeniui atstovaujanti situacija). Situacija $S_{ntp}^{p,kp}$ neegzistuoja p-tosios klasės kp-tajame panašumo lygmenyje, kai tame lygmenyje visos situacijos buvo atpažintos (pagal apskaičiuotus klasterius buvo labiausiai panašios į savo klases). Atstovaujanti situacija $S_{atst}^{p,gp}$ bet kurios p-tosios klasės bet kuriam gp-tajam panašumo lygmeniui yra beveik tokia pati, kaip ir to pačio lygmens neatpažinta situacija $S_{ntp}^{p,gp}$. Jos skiriasi tik tuo, kad $S_{atst}^{p,gp}$ egzistuoja ir tada, kai visos situacijos iš to lygmens jau buvo atpažintos. Tada $S_{atst}^{p,gp}$ būna pati pirmoji situacija iš savo panašumo lygmens, tai yra, situacija, kuri vykdant grupavimą (4.1.6 skyrius 4.8 pav.) buvo parinkta kaip panašumo lygmenį atstovaujanti situacija.

Bendras pirmojo klasterizavimo metodo veikimo principas iliustruojamas nagrinėjant 4.7 paveikslą. Pradžioje kiekvieną panašumo lygmenį atstovaujanti situacija kartu yra ir neatpažinta. Pažymėjus neatpažintas ir atstovaujančias situacijas, gerieji (g) ir blogieji (b) kontraktai pavaizduoti 4.11 paveiksle.



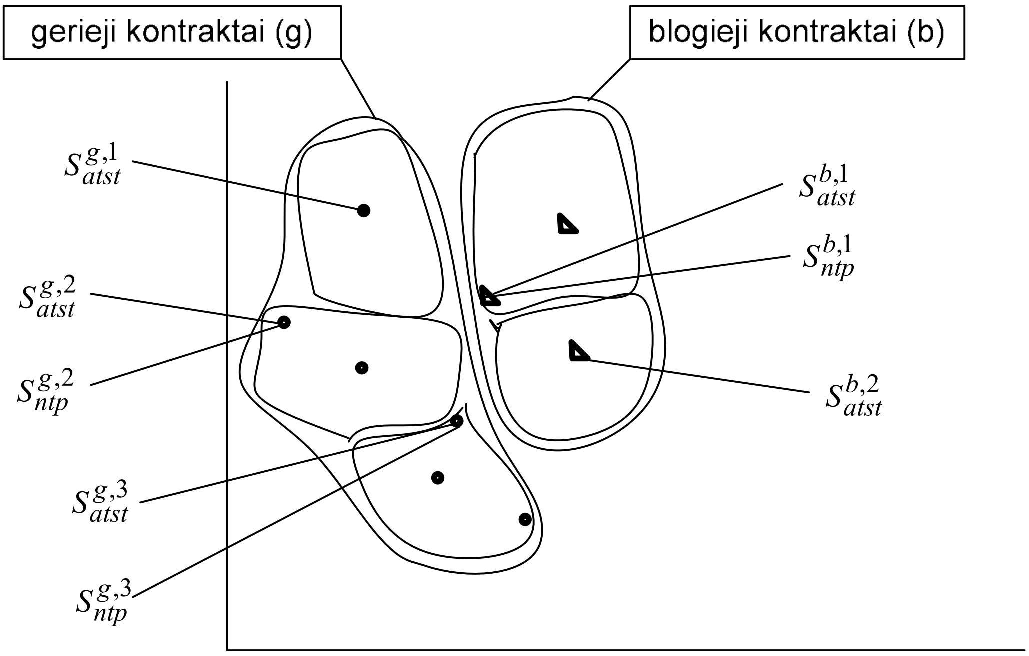
4.11 pav. Gerieji ir blogieji kontraktai

Pritaikius (4.11) formulę šiam atvejui ir skaičiuojant g-tajai klasei pirmąjį klasterį, gaunamas toks uždavinys:

$$\Phi_1^g = \sum_{i=1}^{m} k_{1,i}^g \cdot a_{ntp,i}^{g,1} \to \max$$

$$\begin{cases} \sum_{i=1}^{m} k_{1,i}^g (a_{ntp,i}^{g,1} \cdot \gamma - a_{atst,i}^{g,gg}) \le 0; \qquad \mathrm{gg} = 2, 3; \\ \sum_{i=1}^{m} k_{1,i}^g (a_{atst,i}^{b,gb} - \kappa \cdot a_{ntp,i}^{g,1}) \le 0; \qquad \mathrm{gb} = 1, 2; \\ 0 \le k_{1,i}^g \le D; \qquad \forall i. \end{cases} \tag{4.12}$$

Tokiu pat principu skaičiuojamas klasteris ir b-tajai klasei. Kadangi reikalaujama, kad visos atstovaujančios lygmenis situacijos tarpusavyje būtų γ-panašios, tai dauguma visų tos klasės situacijų bus atpažintos po pirmos iteracijos, tai yra, po to, kai visoms klasėms bus apskaičiuota po pirmąjį pirminių koeficientų klasterį. Po situacijų priskyrimo klasteriams (4.1.6 skyrius 4.9 pav.) 4.11 paveikslas galėtų atrodyti taip (4.12 pav.).



4.12 pav. Kontraktai po pirmojo klasterizavimo metodo pirmosios iteracijos

Visos situacijos buvo atpažintos iš g-tosios klasės pirmojo panašumo lygmens ir b-tosios klasės antrojo panašumo lygmens, todėl tuose lygmenyse liko tik atstovaujančios situacijos, tai yra, pačios pirmosios tų lygmenų situacijos. Kituose lygmenyse neatpažintos ir atstovaujančios situacijos pasikeitė, nes ne visos tų lygmenų situacijos buvo atpažintos. Taip pat tuose lygmenyse per vidurį yra likusios situacijos, kurios jau atpažintos, tačiau pirmojoje iteracijoje jos buvo atstovaujančios tuos lygmenis situacijos. Jei kurio nors lygmens visos situacijos bus atpažintos, tai buvusi atstovaujanti lygmenį situacija vėl taps atstovaujanti. Tokiu principu toliau vykdoma tiek klasterizavimo iteracijų, kol bus atpažintos (priskirtos apskaičiuotiems klasteriams) visos situacijos.

4.2.2. Antrasis klasterizavimo metodas

Antrasis klasterizavimo metodas, atvirkščiai nei pirmasis klasterizavimo metodas (4.2.1 skyrius), realizuotas nesitikint, kad visos situacijos iš skirtingų panašumo lygmenų, yra γ-panašios. Todėl p-tajai klasei TPU formuluojamas (3.3.1 skyrius), paimant pirmą klasteriui dar nepriskirtą situaciją $S_{ntp}^{p,kp}$ iš pirmojo pasitaikiusio panašumo lygmens kp, kuriame dar yra klasteriams nepriskirtų situacijų. Pagal $S_{ntp}^{p,kp}$ sudaroma maksimizuojamoji funkcija. Šiame metode nėra reikalavimo, kad $S_{ntp}^{p,kp}$ su kitom p-tosios klasės panašumo lygmenims atstovaujančiomis situacijomis būtų γ-panašios. Reikalaujama, kad $S_{ntp}^{p,kp}$ su kitų klasių panašumo lygmenims atstovaujančiomis situacijomis būtų κ-nepanašios:

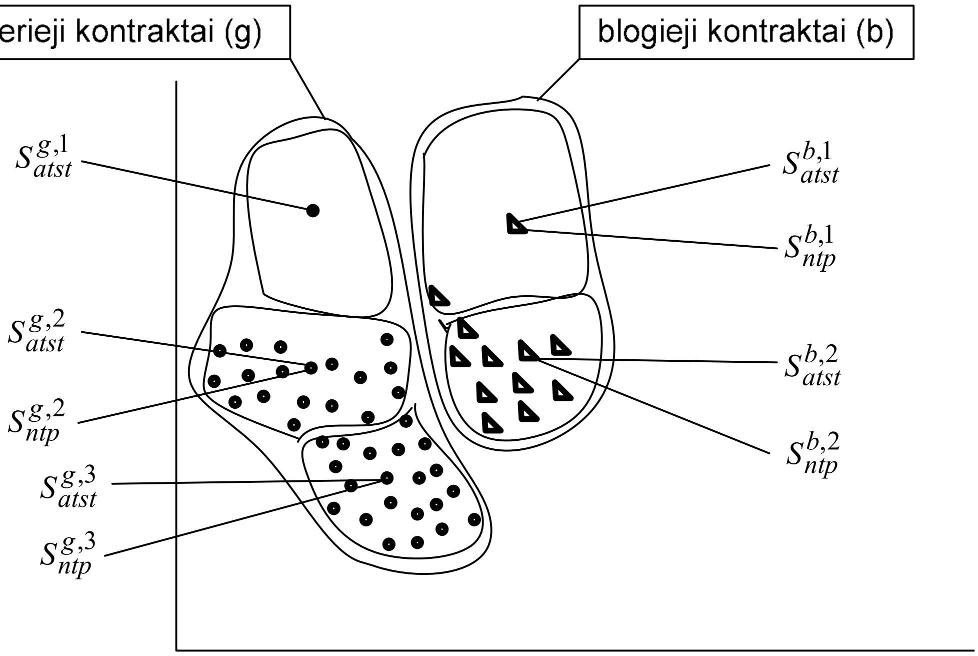
$$\Phi_{wp}^{p} = \sum_{i=1}^{m} k_{wp,i}^{p} \cdot a_{ntp,i}^{p,kp} \rightarrow \max$$
$$\begin{cases} \sum_{i=1}^{m} k_{wp,i}^{p} (a_{atst,i}^{r,gr} - \kappa \cdot a_{ntp,i}^{p,kp}) \leq 0; \quad \mathrm{gr} = 1,...,\mathrm{Gr}; \quad \mathrm{r} = 1,...,\mathrm{z} \neq \mathrm{p} \\ 0 \leq k_{wp,i}^{p} \leq D; \quad \forall i; \quad \mathrm{wp} = 1,...,\mathrm{Wp}. \end{cases} \tag{4.13}$$

TPU formuluojamas p-tajai klasei tik su sąlyga, kad toje klasėje dar yra bent vienas panašumo lygmuo, turintis bent vieną neatpažintą situaciją. Klasterizavimas baigiasi tada, kai nė viena klasė neturi panašumo lygmens, kuriame būtų bent viena neatpažinta situacija. Neatpažintos ir atstovaujančios situacijos apibrėžimai pateikti 4.2.1 skyriuje.

Bendras antrojo klasterizavimo metodo veikimo principas iliustruojamas nagrinėjant 4.11 paveikslą. Pritaikius (4.11) formulę šiam atvejui ir skaičiuojant g-tajai klasei pirmąjį klasterį, gaunamas toks uždavinys:

$$\Phi_{1}^{g} = \sum_{i=1}^{m} k_{1,i}^{g} \cdot a_{ntp,i}^{g,1} \rightarrow \max$$
$$\begin{cases} \sum_{i=1}^{m} k_{1,i}^{g} (a_{atst,i}^{b,gb} - \kappa \cdot a_{ntp,i}^{g,1}) \leq 0; \quad \mathrm{gb} = 1, 2; \\ 0 \leq k_{1,i}^{g} \leq D; \quad \forall i. \end{cases} \tag{4.14}$$

Tokiu pat principu skaičiuojamas klasteris b-tajai klasei. Kadangi nereikalaujama, kad visos atstovaujančios lygmenis situacijos tarpusavyje būtų γ-panašios, tai tik dauguma situacijų, priklausančių tam pačiam panašumo lygmeniui, kaip ir maksimizuojamoji situacija, bus atpažintos po pirmos iteracijos, tai yra, po to, kai visoms klasėms bus apskaičiuota po pirmąjį pirminių koeficientų klasterį. Tikėtina, kad atpažintų (priskirtų klasteriams) situacijų kiekis iš kitų panašumo lygmenų, bus mažesnis. Po situacijų priskyrimo klasteriams (4.1.6 skyrius 4.9 pav.) 4.11 paveikslas galėtų atrodyti taip (4.13 pav.).

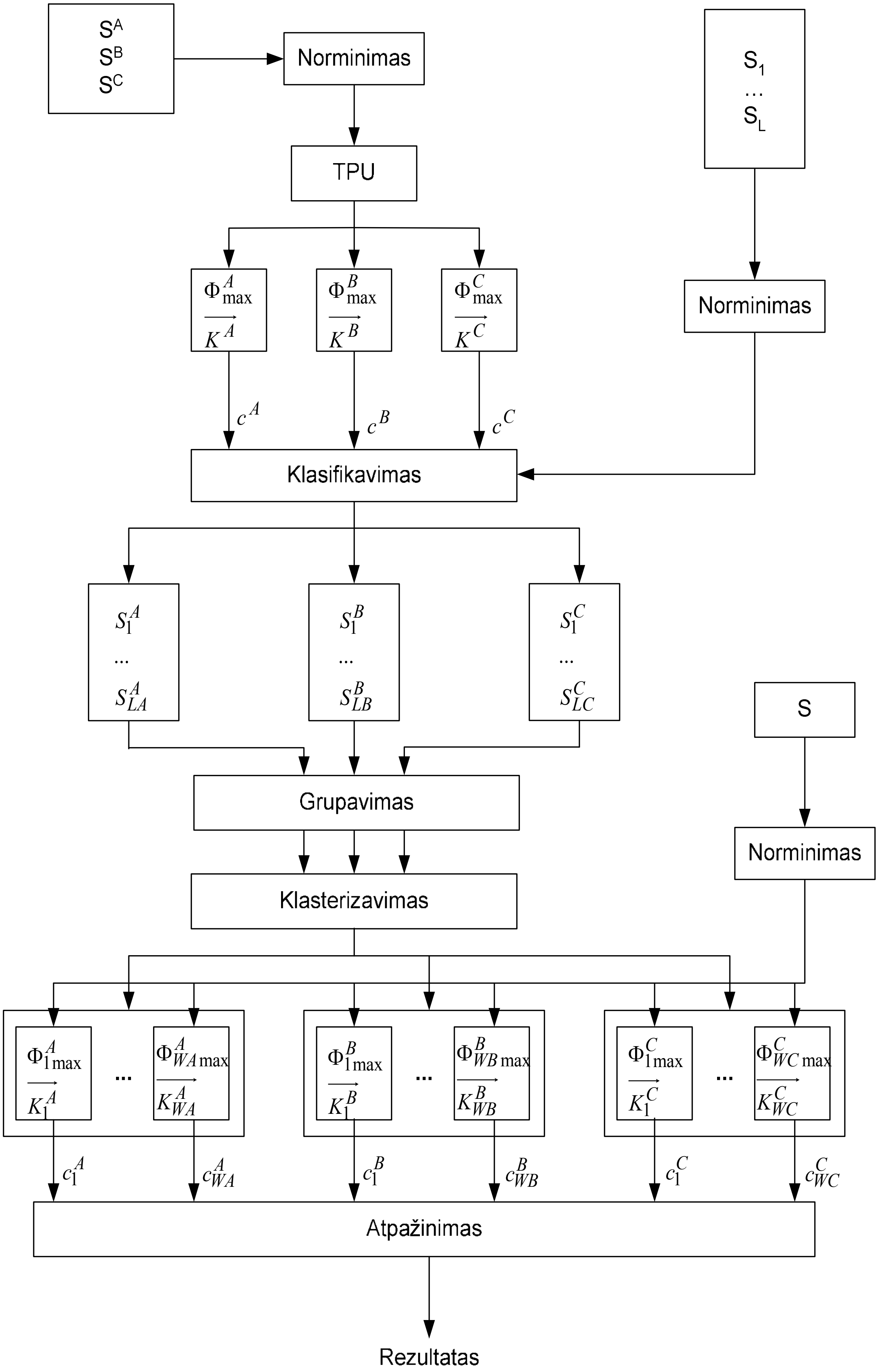


4.13 pav. Kontraktai po antrojo klasterizavimo metodo pirmosios iteracijos

Kadangi g-tosios klasės pirmojo panašumo lygmens visos situacijos buvo atpažintos, todėl tame lygmenyje liko tik atstovaujanti situacija. Klasės b pirmajame panašumo lygmenyje neatpažinta ir atstovaujanti situacija pasikeitė, nes ne visos to lygmens situacijos buvo atpažintos. Taip pat tame lygmenyje per vidurį yra likusi situacija, kuri jau atpažinta, tačiau pirmojoje iteracijoje ji buvo atstovaujanti tą lygmenį situacija. Jei šio lygmens visos situacijos bus atpažintos, tai buvusi atstovaujanti lygmenį situacija vėl taps atstovaujanti. Tokiu principu toliau vykdoma tiek klasterizavimo iteracijų, kol bus atpažintos (priskirtos apskaičiuotiems klasteriams) visos situacijos.

4.3. Bendra miglotai apibrėžtos situacijos įvertinimo modulio schema

Apibendrinus visas išdėstytas situacijų norminimo, klasifikavimo, grupavimo, klasterizavimo, atpažinimo procedūras ir jų sąveiką, sudaryta bendra miglotai apibrėžtos situacijos įvertinimo modulio veiksmų schema (4.14 pav.). Šioje schemoje pavaizduotos dar neklasifikuotos situacijos, kiekvienos klasės išreikštinių parametrų tipiniai atstovai ir nauja situacija, kuri dar bus atpažinta. Visos šios situacijos norminamos (4.1.4 skyrius). Taip pat schemoje matosi pagrindiniai pirminių parametrų reikšmingumo koeficientų radimo etapai: klasifikavimas, grupavimas, klasterizavimas (4.1.6 skyrius). Be to, įtrauktas ir naujos situacijos atpažinimas (4.1.7 skyrius) bei maksimizuojamų funkcijų norminimas (3.12).



4.14 pav. Bendra miglotai apibrėžtos situacijos įvertinimo modulio veiksmų schema

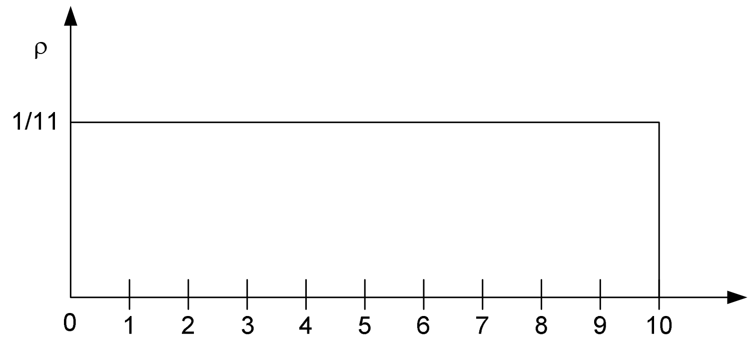
4.4. Pirminiai tyrimai

Norint apskritai ištirti miglotai apibrėžtos situacijos modulio sukūrimo galimybes bei veikimo principus, pirmajame tyrimų etape buvo pasirinkti abstraktūs, nesusiję su realiais kontraktais, duomenys. Pirminiai duomenys buvo sugeneruoti dirbtinai iš tolygiai atsitiktinių skaičių aibės. Pirmiausiai buvo generuojami situacijų išreikštiniai parametrai. Tada toms situacijoms reikėjo kokiu nors būdu sugeneruoti ir pirminius parametrus taip, kad kiekvienos situacijos pirminiai parametrai būtų bent statistiškai susiję su išreikštiniais parametrais (4.1), nes būtent pagal pastaruosius vykdomas situacijų klasifikavimas. Taip pat reikėjo sukurti ar parinkti kiekvienos klasės išreikštinių parametrų tipinius atstovus (4.2), pagal kuriuos skaičiuojami išreikštiniai koeficientai ir paskui atliekamas klasifikavimas (plačiau apie situacijų klasifikavimą 4.1.6 skyrius 4.6 pav.). Tyrimuose tiek pirminių, tiek išreikštinių parametrų įverčiai buvo generuojami iš sveikų skaičių intervalo [0,10]. Parinktas koeficientų reikšmių kitimo intervalas – [0,10], tai yra, (3.14) formulėje D reikšmė nustatyta 10. Rezultatų apžvelgimui palengvinti situacijų pirminių parametrų kiekis buvo pasirinktas 6, o išreikštinių parametrų – 4. Bendras sugeneruotų situacijų kiekis – 100. Atsitiktinai pasirinktos trys klasės, kurios atitinkamai pavadintos A, B ir C klasėmis.

Su dirbtinai generuotais duomenimis buvo bandomi abu klasterizavimo metodai. Klasterizuojant pirmuoju metodu (4.2.1 skyrius), pirminiai koeficientai gavosi nuliniai, nes formuluojant tiesinio programavimo uždavinį reikalaujama, kad vienos klasės situacijos iš skirtingų panašumo lygmenų būtų tarpusavyje γ-panašios. Šis reikalavimas atsitiktinai generuotiems duomenims yra per griežtas, nes vienoje klasėje gali būti labai skirtingų situacijų. Su dirbtiniais duomenimis pavyko antrasis klasterizavimo metodas (4.2.2 skyrius), kurio skaičiavimų rezultatai 4.4.2 skyriuje.

4.4.1. Situacijų parametrų įverčių generavimas

Pirmiausia atsitiktinai generuojami šimto situacijų išreikštinių parametrų įverčiai iš vienodo pasiskirstymo tankio (4.15 pav.).



4.15 Pav. Išreikštinių parametrų pasiskirstymo tankiai

Tada tyrimo patogumui dirbtinai nustatoma kiekvienai klasei po vieną išreikštinių parametrų tipinį atstovą (4.2). Tie tipiniai atstovai pateikiami 4.1 lentelėje.

4.1 lentelė. Klasių tipiniai atstovai

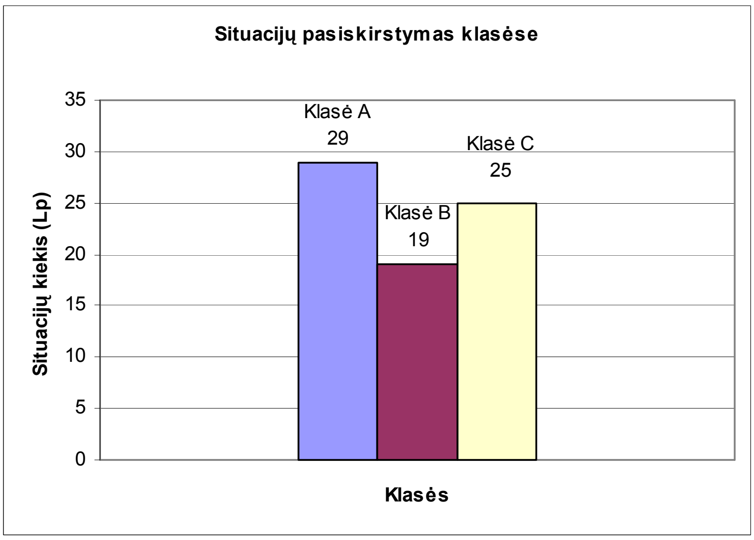
Tipiniai atstovai	Klasė A	Klasė B	Klasė C
	T^A_1	T^B_1	T^C_1
b_1	6.00	7.00	4.00
b_2	0.00	10.00	4.00
b_3	9.00	3.00	7.00
b_4	0.00	9.00	4.00

Šie išreikštinių parametrų tipiniai atstovai centruojami (3.5), ir pagal juos apskaičiuojami išreikštiniai koeficientai ir maksimizuojamos funkcijos (4.3), kurios norminamos pagal apskaičiuotas konstantas (3.12). Tie išreikštiniai koeficientai, maksimizuojamos funkcijos ir jų norminimo konstantos pateikiami 4.2 lentelėje.

4.2 lentelė. Dirbtinių duomenų išreikštinių koeficientų klasteriai

Išreikštiniai koeficientai	Klasė A	Klasė B	Klasė C
	K^A	K^B	K^C
k_1	10.00	0.00	0.00
k_2	0.00	10.00	6.59
k_3	10.00	0.00	10.00
k_4	0.00	10.00	6.74
Max funkcijos	Φ^A_{max}	Φ^B_{max}	Φ^C_{max}
	75.00	45.00	12.50
Konstantos	c^A	c^B	c^C
	1.00	1.67	6.00

Paskui pagal gautus koeficientus situacijos klasifikuojamos (4.1.6 skyrius 4.6 pav.). Klasifikuojant dvidešimt septynios iš šimto situacijų nebuvo panašios nei į vieną klasę, nes klasių tipiniai atstovai neaprėpė visų sugeneruotų išreikštinių parametrų variantų. Todėl liko 73 suklasifikuotos situacijos. Situacijų pasiskirstymas į klases parodytas 4.16 paveiksle.



4.16 pav. Dirbtinių situacijų pasiskirstymas klasėse

Po klasifikavimo apskaičiuojama visų sugeneruotų išreikštinių parametrų įverčių iš intervalo [0,10] pasiskirstymo dėsniai ir tankiai kiekvienai klasei atskirai (4.3, 4.4, 4.5 lentelės).

4.3 lentelė. A klasės išreikštinių parametrų įverčių pasiskirstymo dėsnis ir tankis

Skaičius	0	1	2	3	4	5	6	7	8	9	10
dėsnis	0.0900	0.1803	0.2737	0.3686	0.4639	0.5569	0.6482	0.7372	0.8236	0.9109	1.0000
tankis	0.0900	0.0903	0.0935	0.0948	0.0953	0.0930	0.0913	0.0890	0.0864	0.0873	0.0891

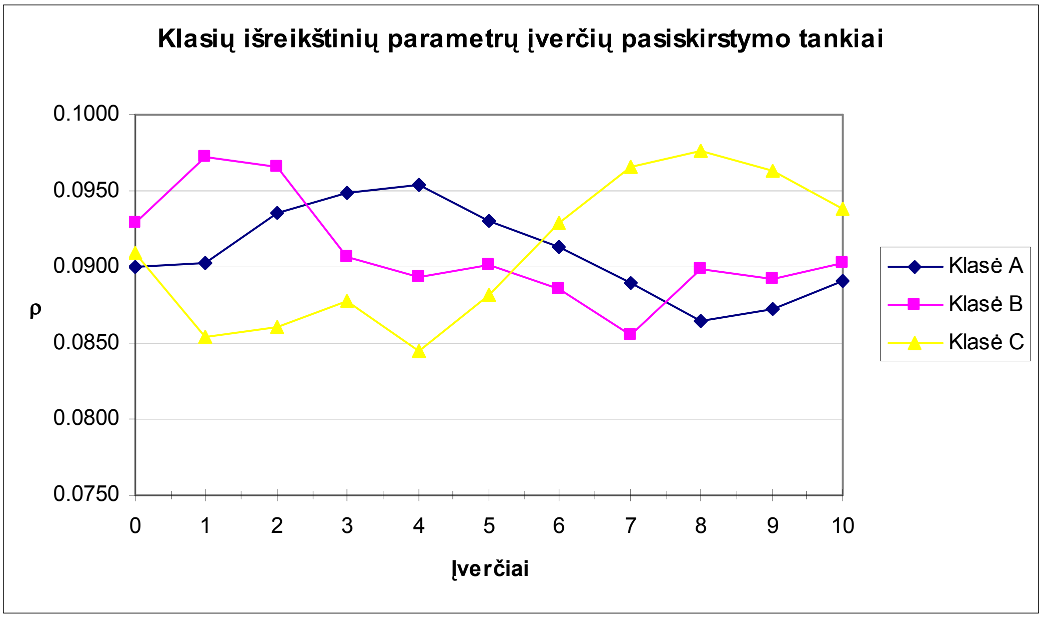
4.4 lentelė. B klasės išreikštinių parametrų įverčių pasiskirstymo dėsnis ir tankis

Skaičius	0	1	2	3	4	5	6	7	8	9	10
dėsnis	0.0929	0.1901	0.2867	0.3773	0.4667	0.5567	0.6452	0.7308	0.8206	0.9098	1.0000
tankis	0.0929	0.0972	0.0965	0.0907	0.0893	0.0901	0.0885	0.0855	0.0898	0.0891	0.0902

4.5 lentelė. C klasės išreikštinių parametrų įverčių pasiskirstymo dėsnis ir tankis

Skaičius	0	1	2	3	4	5	6	7	8	9	10
dėsnis	0.0909	0.1763	0.2624	0.3501	0.4346	0.5227	0.6156	0.7122	0.8098	0.9061	1.0000
tankis	0.0909	0.0854	0.0861	0.0877	0.0845	0.0881	0.0930	0.0966	0.0976	0.0963	0.0939

Išreikštinių parametrų įverčių pasiskirstymo tankiai pavaizduoti 4.17 paveiksle.



4.17 pav. Išreikštinių parametrų įverčių pasiskirstymo tankiai

Turint visų situacijų išreikštinių parametrų įverčius, reikia toms situacijoms sugeneruoti ir pirminių parametrų įverčius taip, kad kiekvienos situacijos pirminių ir išreikštinių parametrų įverčiai būtų bent statistiškai susiję. Todėl buvo apskaičiuoti išreikštinių parametrų įverčių pasiskirstymo dėsniai ir tankiai kiekvienai klasei atskirai. Visų situacijų pirminių parametrų įverčiai kiekvienai klasei generuojami pagal tos klasės išreikštinių parametrų įverčių pasiskirstymo dėsnius. Iš pradžių sugeneruojamas atsitiktinis realus skaičius iš intervalo [0,1]. Tada nustatoma sugeneruoto realaus skaičiaus tikrasis įvertis pagal tos klasės išreikštinių parametrų pasiskirstymo dėsnį, tikrinant, į kurį pasiskirstymo dėsnio skaičių intervalą sugeneruotasis skaičius patenka. Pvz., jei klasei A būtų sugeneruotas realus skaičius 0,15, tai jis patektų į pasiskirstymo dėsnio intervalą (0,0900;0,1803], ir jam būtų suteiktas įvertis 1; realiam skaičiui 0,39 – įvertis 4 ir t.t.

Pagal sugeneruotus pirminių parametrų įverčius apskaičiuojami jų pasiskirstymo tankiai bei dėsniai kiekvienai klasei (4.6, 4.7, 4.8 lentelės).

4.6 lentelė. A klasės pirminių parametrų įverčių pasiskirstymo dėsnis ir tankis

Skaičius	0	1	2	3	4	5	6	7	8	9	10
dėsnis	0.0900	0.1804	0.2738	0.3691	0.4635	0.5562	0.6495	0.7376	0.8241	0.9112	1.0000
tankis	0.0900	0.0905	0.0934	0.0953	0.0945	0.0926	0.0933	0.0881	0.0866	0.0871	0.0888

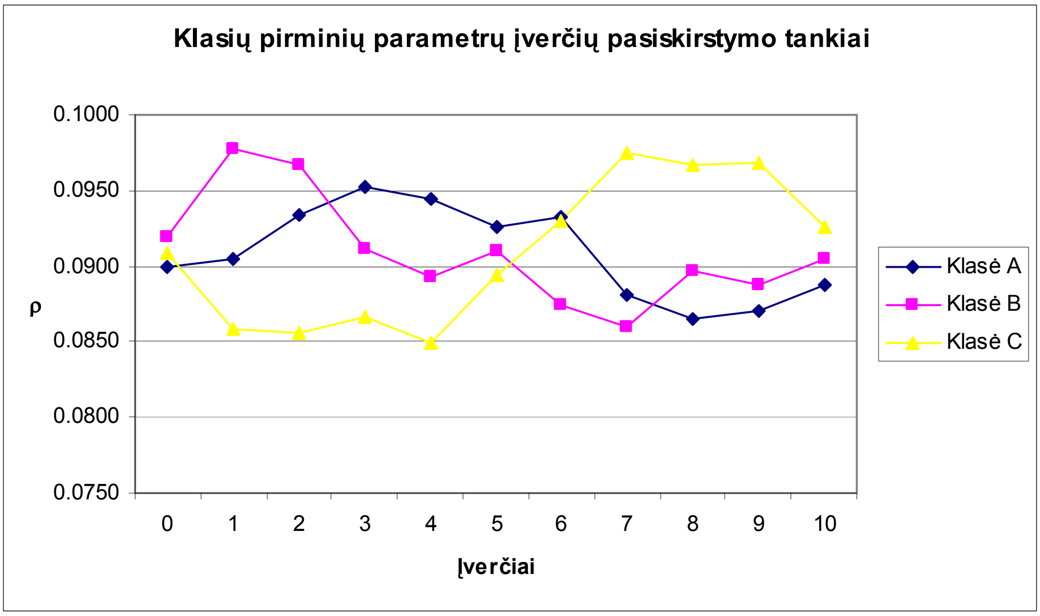
4.7 lentelė. B klasės pirminių parametrų įverčių pasiskirstymo dėsnis ir tankis

Skaičius	0	1	2	3	4	5	6	7	8	9	10
dėsnis	0.0919	0.1897	0.2863	0.3774	0.4667	0.5577	0.6451	0.7311	0.8207	0.9095	1.0000
tankis	0.0919	0.0978	0.0966	0.0911	0.0893	0.0910	0.0874	0.0860	0.0897	0.0888	0.0905

4.8 lentelė. C klasės pirminių parametrų įverčių pasiskirstymo dėsnis ir tankis

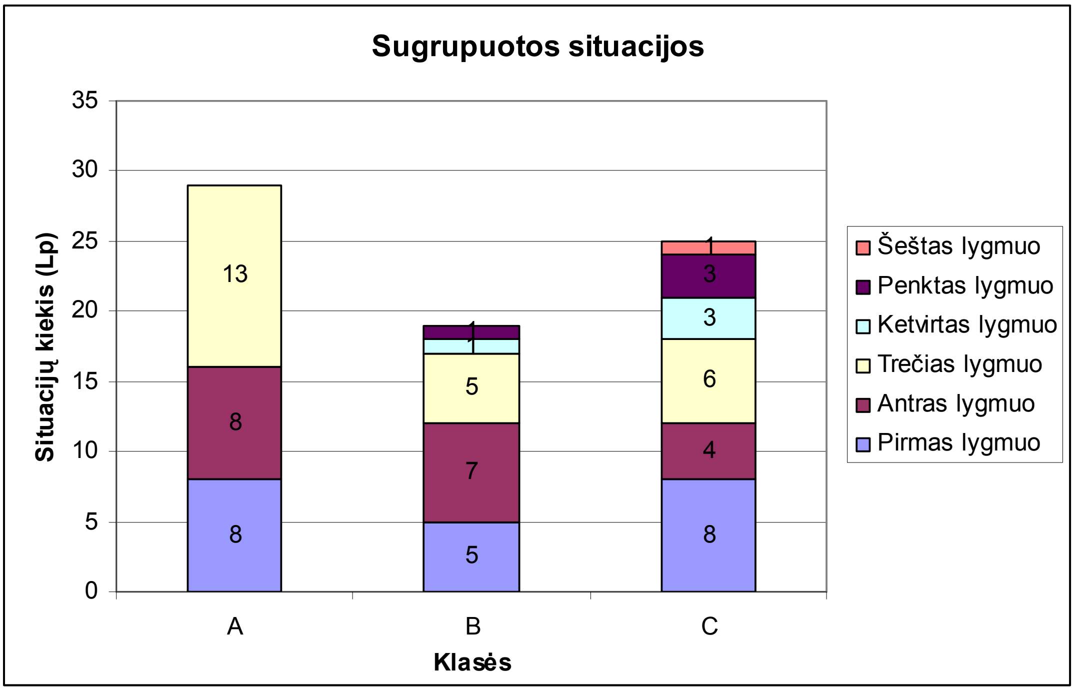
Skaičius	0	1	2	3	4	5	6	7	8	9	10
dėsnis	0.0908	0.1767	0.2623	0.3490	0.4340	0.5234	0.6164	0.7138	0.8106	0.9074	1.0000
tankis	0.0908	0.0859	0.0856	0.0867	0.0849	0.0894	0.0930	0.0974	0.0967	0.0968	0.0926

Pirminių parametrų įverčių pasiskirstymo tankiai pavaizduoti 4.18 paveiksle.



4.18 pav. Pirminių parametrų įverčių pasiskirstymo tankiai

Toliau jau sugeneruotos situacijos grupuojamos į panašumo lygmenis pagal pirminių parametrų įverčius (4.7). Atlikus tokį grupavimą, klasės A situacijos pasiskirstė į tris panašumo lygmenis, klasės B – į penkis, klasės C – į šešis (4.19 pav.).



4.19 pav. Sugrupuotos dirbtinės situacijos klasėse

4.4.2. Dirbtinių situacijų koeficientų radimas antruoju klasterizavimo metodu

Skaičiuojant dirbtinių duomenų pirminius koeficientus antruoju klasterizavimo metodu, klasei A buvo suskaičiuoti trys klasteriai, klasei B – 2, klasei C – 5. Šių klasių visų klasterių koeficientai, maksimizavimo funkcijos ir jų norminimo konstantos parodyti 4.9 lentelėje.

4.9 lentelė. Dirbtinių situacijų pirminių koeficientų klasteriai

Pirminiai koeficientai	Klasė A			Klasė B		Klasė C				
	K^A_1	K^A_2	K^A_3	K^B_1	K^B_2	K^C_1	K^C_2	K^C_3	K^C_4	K^C_5
k_1	10.00	5.19	10.00	0.00	1.49	0.00	0.00	10.00	10.00	7.96
k_2	9.50	3.67	10.00	0.00	6.86	10.00	10.00	0.00	0.00	0.00
k_3	0.00	10.00	10.00	3.54	0.00	0.00	4.64	0.00	2.61	10.00
k_4	0.00	5.46	0.00	3.76	0.37	0.00	1.82	10.00	10.00	0.00
k_5	0.00	0.14	0.00	2.18	10.00	10.00	1.55	0.00	3.39	8.25
k_6	0.00	0.00	5.80	10.00	8.17	8.11	3.85	6.56	2.63	1.72
Max funkcijos	Φ^A_{1max}	Φ^A_{2max}	Φ^A_{3max}	Φ^B_{1max}	Φ^B_{2max}	Φ^C_{1max}	Φ^C_{2max}	Φ^C_{3max}	Φ^C_{4max}	Φ^C_{5max}
	85.00	19.71	55.46	44.56	6.90	73.78	24.01	97.71	52.13	28.47
Konstantos	c^A_1	c^A_2	c^A_3	c^B_1	c^B_2	c^C_1	c^C_2	c^C_3	c^C_4	c^C_5
	1.15	4.96	1.76	2.19	14.16	1.32	4.07	1.00	1.87	3.43

Lentelėje 4.10 pateikta po vieną situaciją iš apskaičiuotų klasterių, parodytų 4.9 lentelėje.

4.10 lentelė. Dirbtinių situacijų pirminiai parametrai

Parametrai	Klasė A			Klasė B		Klasė C				
	S^A_1	S^A_2	S^A_3	S^B_1	S^B_2	S^C_1	S^C_2	S^C_3	S^C_4	S^C_5
a_1	9.00	2.00	6.00	4.00	3.00	3.00	5.00	7.00	10.00	3.00
a_2	7.00	4.00	7.00	4.00	7.00	7.00	10.00	0.00	3.00	3.00
a_3	1.00	7.00	7.00	7.00	3.00	3.00	5.00	0.00	5.00	6.00
a_4	1.00	7.00	3.00	6.00	9.00	9.00	8.00	9.00	8.00	7.00
a_5	3.00	4.00	1.00	2.00	9.00	8.00	6.00	0.00	6.00	10.00
a_6	1.00	4.00	8.00	10.00	0.00	0.00	4.00	4.00	4.00	1.00

Rezultatų, pateiktų 4.9 ir 4.10 lentelėse, vaizdesniam pateikimui celės nuspalvinamos tuo tamsesne spalva, kuo didesnė yra reikšmingumo koeficiento reikšmė toje celėje, pavyzdžiui, balta spalva apima intervalą [0; 3,33], pilka – (3,33; 6,66], juoda (6,66; 10]. Taip apdoroti rezultatai pateikti 4.11 ir 4.12 lentelėse.

4.11 lentelė. Dirbtinių situacijų pirminių koeficientų reikšmingumas

Pirminiai koeficientai	Klasė A			Klasė B		Klasė C				
	K^A_1	K^A_2	K^A_3	K^B_1	K^B_2	K^C_1	K^C_2	K^C_3	K^C_4	K^C_5
k_1										
k_2										
k_3										
k_4										
k_5										
k_6										

4.12 lentelė. Dirbtinių situacijų pirminių parametrų reikšmingumas

Parametrai	Klasė A			Klasė B		Klasė C				
	S^A_1	S^A_2	S^A_3	S^B_1	S^B_2	S^C_1	S^C_2	S^C_3	S^C_4	S^C_5
a_1										
a_2										
a_3										
a_4										
a_5										
a_6										

Iš 4.11 lentelės matosi, kad klasei A didžiausią reikšmę turi koeficientai k_1, k_2 ir k_3; klasei B – k_2, k_5, k_6; klasei C – k_1 ir k_4 arba k_1, k_3 ir k_5. Taip pat B-tosios klasės antrasis klasteris K^B_2 ir C-tosios klasės pirmasis klasteris K^C_1 yra labai panašūs, nes skirtingose klasėse buvo sugeneruota labai panašių situacijų.

4.4.3. Dirbtinių situacijų atpažinimas

Suklasterizavus dirbtines situacijas, buvo išbandytas tų pačių situacijų atpažinimas pagal apskaičiuotus pirminius koeficientus (4.1.7 skyrius). Penkiolika iš visų 73-jų situacijų buvo atpažintos neteisingai, tai yra, priskirtos ne tai klasei, kuriai iš tikrųjų priklauso. Keleto neteisingai atpažintų situacijų pirminių parametrų įverčiai pateikti 4.13 lentelėje.

4.13 lentelė. Neteisingai atpažintų situacijų pirminiai parametrai

Parametrai	Situacijos					
	A→B	A→B	A→B	C→B	C→B	C→B
a_1	10.00	8.00	9.00	0.00	1.00	6.00
a_2	6.00	9.00	8.00	10.00	10.00	6.00
a_3	0.00	1.00	4.00	0.00	5.00	5.00
a_4	0.00	3.00	2.00	5.00	7.00	8.00
a_5	5.00	3.00	7.00	7.00	9.00	9.00
a_6	3.00	4.00	6.00	2.00	0.00	5.00

Čia žymėjimas „A→B“ reiškia, kad situacija, priklausanti klasei A, buvo atpažinta ir priskirta klasei B. Šios neteisingai atpažintos situacijos vaizdžiau pateiktos 4.14 lentelėje.

4.14 lentelė. Neteisingai atpažintų situacijų pirminių parametrų reikšmingumas

Parametrai	Situacijos					
	A→B	A→B	A→B	C→B	C→B	C→B
a_1						
a_2						
a_3						
a_4						
a_5						
a_6						

4.5. Tyrimai su realiais duomenimis

Šio darbo tyrimui realūs kontraktai buvo paimti iš vienos Vokietijos kredito kompanijos, todėl parametrų pavadinimai pateikti vokiečių kalba. Buvo gauti 343 patyriminiai kontraktai. Kiekvieną kontraktą sudaro 51 pirminis parametras. Buvo nustatytos kiekvieno parametro baigtinės ribos (3.2), ir jie buvo sunorminti (3.1 skyrius) intervale [0,10]. Parinktas koeficientų reikšmių kitimo intervalas – [0,10], tai yra, (3.14) formulėje D reikšmė nustatyta 10. Visi patyriminiai kontraktai jau klasifikuoti pagal tos kompanijos vidinę kontraktų vertinimo sistemą. Kontraktai turi tik vieną išreikštinį parametrą, kuris iš karto nusako, ar tai geras, ar blogas, ar toliau nagrinėtinas kontraktas. Kredito kompanijoje kiekvienas kontraktas vertinamas tam tikra nustatyta tvarka. Jeigu kontrakto parametrų įverčiai geri, tai jis iš karto žymimas kaip priimtas kontraktas, jei įverčiai blogi – atmestas kontraktas, o jei įverčiai vidutiniai arba tam tikri parametrai neužpildyti (neturi įverčių), tai kontraktas pažymimas, kad jį reikia įvertinti rankiniu būdu. Todėl buvo išskirtos trys klasės:

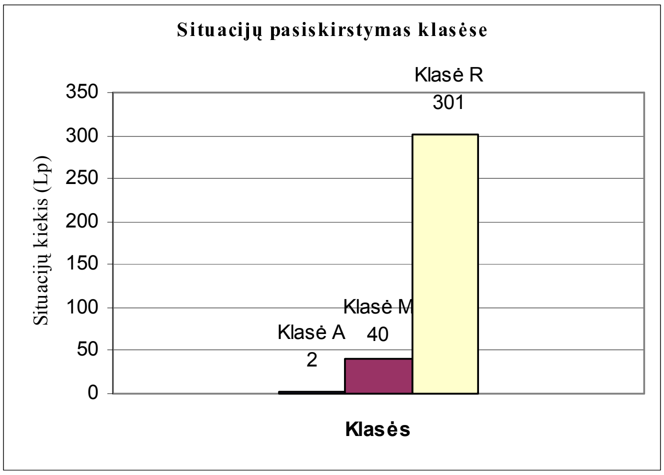
A – priimti (angl. *Accepted*), M – automatiškai neįvertinti (angl. *Manual*), R – atmesti (angl. *Rejected*) kontraktai. Toliau šios klasės bus žymimos pirmosiomis jų angliško vertimo raidėmis.

Pirminių parametrų pavadinimai ir numeracija pateikta 4.15 lentelėje.

4.15 lentelė. Realių kontraktų pirminių parametrų pavadinimai

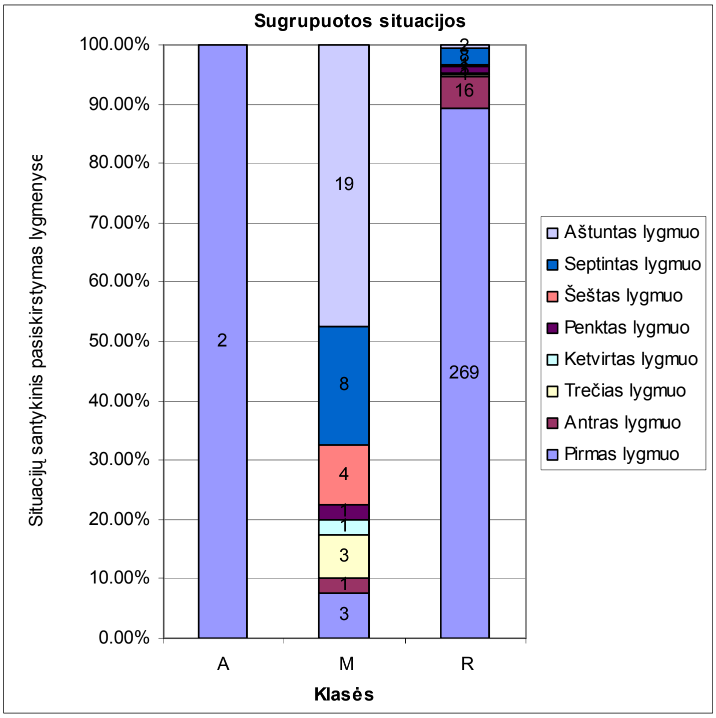
a_{1}	IN V UMFINANZIERUNG
a_{2}	IN V MBG
a_{3}	IN V RESTWERT
a_{4}	IN V BALLONRATE
a_{5}	IN V MSZ
a_{6}	IN V VERTRAGSART
a_{7}	IN V LAUFZEIT
a_{8}	IN V RATE
a_{9}	IN V VERTRAGSZINS
a_{10}	IN O AUSLANDSEINSATZ
a_{11}	IN O MODUL
a_{12}	IN O ALTER
a_{13}	IN O ZUSTAND
a_{14}	IN K BESTAND
a_{15}	IN K ADRESSTYP
a_{16}	IN K RECHTSFORM
a_{17}	IN K LAND
a_{18}	IN K WATCHLIST
a_{19}	IN K ABGELEHNTE ANTRAEGE
a_{20}	IN K OFFENE ANFRAGEN
a_{21}	IN K OFFENER POSTEN
a_{22}	IN K ANZAHL RUELA 6M
a_{23}	IN K ANZAHL RUELA 12M
a_{24}	IN K ANZAHL RUELA 18M
a_{25}	IN K MAX ZUL OP
a_{26}	IN K OBLIGO
a_{27}	IN K STUNDUNGEN
a_{28}	IN K UNTERN ALTER
a_{29}	IN K BRANCHE
a_{30}	IN K BRANCHE GRUPPE
a_{31}	IN K VC INDEX
a_{32}	IN K VC ZAHLUNGSWEISE
a_{33}	IN K VC STAMMKAPITAL
a_{34}	IN K SCHUFA KLASSE
a_{35}	IN K SCHUFA SCORE
a_{36}	IN K ALTER
a_{37}	IN K BESCHAEFTIGUNGSVERHAELTNIS
a_{38}	IN K BESCHAEFTIGUNGSDAUER
a_{39}	IN K JAHRESEINKOMMEN
a_{40}	IN K ZUSATZEINKOMMEN
a_{41}	IN K JAHRESUEBERSCHUSS
a_{42}	IN K BERUFSGRUPPE
a_{43}	IN VM GEWERBLICH
a_{44}	IN VM PRIVAT
a_{45}	IN VM KOOPERATION MIT
a_{46}	IN VM VC INDEX
a_{47}	IN VM WATCHLIST
a_{48}	IN VM GEKUENDIGTE VERTRAEGE
a_{49}	IN VM VERMITTELTE VERTRAEGE
a_{50}	IN VM DAUER GESCHBEZIEHUNG
a_{51}	IN VM LAND

Visi 343 kontraktai buvo suklasifikuoti pagal vienintelį išreikštinį parametrą. Klasei A priklauso dvi situacijos, klasei B – 40, klasei C – 301 (4.20 pav.).



4.20 pav. Realių situacijų pasiskirstymas klasėse

Toliau situacijos grupuojamos į panašumo lygmenis pagal pirminių parametrų įverčius (4.7). Atlikus grupavimą, klasės A situacijos patenka į vieną panašumo lygmenį, klasės B – į aštuonis, klasės C – taip pat į aštuonis (4.21 pav.).



4.21 pav. Sugrupuotos realios situacijos klasėse

4.5.1. Realių situacijų koeficientų radimas pirmuoju klasterizavimo metodu

Skaičiuojant realių duomenų pirminius koeficientus pirmuoju klasterizavimo metodu, klasei A buvo suskaičiuotas vienas klasteris, klasei M – 2, klasei R – 2. Šių klasių visų klasterių koeficientai, maksimizavimo funkcijos ir jų norminimo konstantos parodyti 4.16 lentelėje.

4.16 lentelė. Realių situacijų pirminių koeficientų klasteriai

Pirminiai koeficientai	Klasė A	Klasė M		Klasė R	
	K^A_1	K^M_1	K^M_2	K^R_1	K^R_2
k_1	10.00	7.39	10.00	10.00	10.00
k_2	10.00	0.00	0.00	10.00	0.00
k_3	0.00	6.25	10.00	2.07	4.77
k_4	3.36	0.00	8.33	10.00	10.00
k_5	10.00	0.96	0.00	0.00	10.00
k_6	0.00	1.89	2.84	10.00	9.42
k_7	10.00	5.00	9.41	2.92	0.00
k_8	10.00	0.00	10.00	10.00	0.00
k_9	0.00	10.00	10.00	0.00	0.00
k_{10}	6.57	10.00	10.00	0.00	0.00
k_{11}	0.00	1.91	3.02	6.35	7.54
k_{12}	10.00	0.00	0.00	10.00	0.00
k_{13}	10.00	0.00	3.96	2.20	1.02
k_{14}	10.00	10.00	10.00	10.00	7.74
k_{15}	10.00	10.00	10.00	0.00	0.00
k_{16}	0.00	10.00	10.00	0.00	0.00
k_{17}	8.67	10.00	10.00	0.00	0.00
k_{18}	4.28	10.00	10.00	0.00	0.00
k_{19}	3.16	3.45	9.95	0.56	10.00
k_{20}	4.72	2.14	6.61	10.00	0.57
k_{21}	3.68	3.05	4.22	4.39	3.75
k_{22}	10.00	10.00	10.00	10.00	0.00
k_{23}	1.32	1.03	4.04	1.21	0.00
k_{24}	0.00	0.00	0.00	0.00	0.00
k_{25}	9.33	6.54	10.00	10.00	7.40
k_{26}	0.00	0.00	0.66	2.14	0.00
k_{27}	0.00	0.00	0.00	0.00	0.00
k_{28}	8.66	10.00	7.95	0.00	0.00
k_{29}	0.00	0.00	0.00	10.00	10.00
k_{30}	0.00	0.00	0.00	6.19	2.40
k_{31}	0.00	0.00	0.00	10.00	10.00
k_{32}	10.00	1.15	4.77	9.66	0.11
k_{33}	10.00	10.00	10.00	0.00	0.00
k_{34}	0.00	0.00	0.00	10.00	10.00
k_{35}	4.69	10.00	10.00	0.00	0.00
k_{36}	0.00	9.67	10.00	5.55	0.00
k_{37}	6.14	4.74	0.00	3.73	0.00
k_{38}	9.01	0.28	3.91	5.23	0.35
k_{39}	5.75	10.00	10.00	0.00	0.00
k_{40}	0.00	0.00	0.00	0.00	0.00
k_{41}	0.00	0.00	0.00	0.00	0.00
k_{42}	0.00	0.00	0.00	0.00	0.00
k_{43}	1.17	5.27	5.41	4.29	5.87
k_{44}	9.92	10.00	10.00	6.07	2.82
k_{45}	0.00	0.00	0.00	0.00	0.00
k_{46}	3.88	10.00	10.00	0.00	0.00
k_{47}	0.00	0.00	0.00	0.00	0.00
k_{48}	3.93	3.52	4.99	3.89	2.87
k_{49}	4.71	4.82	5.84	3.71	2.34
k_{50}	0.00	0.72	3.84	0.00	0.00
k_{51}	2.79	1.64	9.84	3.50	4.18
Max funkcijos	Φ^A_{1max}	Φ^M_{1max}	Φ^M_{2max}	Φ^R_{1max}	Φ^R_{2max}
	171.16	82.51	153.80	81.95	74.09
Konstantos	c^A_1	c^M_1	c^M_2	c^R_1	c^R_2
	1.00	2.07	1.11	2.09	2.31

Apskaičiuoti pirminiai koeficientai vaizdžiau pateikti 4.17 lentelėje.

4.17 lentelė. Realių situacijų pirminių koeficientų reikšmingumas

Pirminiai koeficientai	Klasė A	Klasė M		Klasė R	
	K^A_1	K^M_1	K^M_2	K^R_1	K^R_2
k_1					
k_2					
k_3					
k_4					
k_5					
k_6					
k_7					
k_8					
k_9					
k_{10}					
k_{11}					
k_{12}					
k_{13}					
k_{14}					
k_{15}					
k_{16}					
k_{17}					
k_{18}					
k_{19}					
k_{20}					
k_{21}					
k_{22}					
k_{23}					
k_{24}					
k_{25}					
k_{26}					
k_{27}					
k_{28}					
k_{29}					
k_{30}					
k_{31}					
k_{32}					
k_{33}					
k_{34}					
k_{35}					
k_{36}					
k_{37}					
k_{38}					
k_{39}					
k_{40}					
k_{41}					
k_{42}					
k_{43}					
k_{44}					
k_{45}					
k_{46}					
k_{47}					
k_{48}					
k_{49}					
k_{50}					
k_{51}					

Lentelėje 4.18 pateikta po vieną situaciją iš apskaičiuotų klasterių, parodytų 4.16 lentelėje.

4.18 lentelė. Realių situacijų pirminiai parametrai

Parametrai	Klasė A	Klasė M		Klasė R	
	S^A_1	S^M_1	S^M_2	S^R_1	S^R_2
a_1	0.00	0.00	0.00	0.00	0.00
a_2	4.99	2.78	1.00	2.32	1.00
a_3	0.00	3.71	0.00	0.00	0.00
a_4	0.00	0.00	0.00	0.00	0.00
a_5	4.99	1.85	0.50	0.00	0.33
a_6	2.50	5.00	2.50	2.50	2.50
a_7	8.33	5.00	1.67	1.67	1.67
a_8	0.73	0.42	0.74	2.02	0.78
a_9	0.00	0.00	0.00	0.00	0.00
a_{10}	0.00	0.00	0.00	0.00	0.00
a_{11}	0.39	0.49	0.15	0.19	0.15
a_{12}	1.82	4.55	0.00	0.00	0.00
a_{13}	2.50	5.00	2.50	2.50	2.50
a_{14}	0.00	0.00	0.00	0.00	0.00
a_{15}	5.00	2.50	5.00	2.50	2.50
a_{16}	0.25	0.75	0.25	0.00	0.00
a_{17}	2.05	2.05	2.05	2.05	2.05
a_{18}	0.00	5.00	0.00	0.00	0.00
a_{19}	0.00	0.00	2.00	0.00	0.00
a_{20}	0.00	0.00	10.00	0.00	0.00
a_{21}	0.00	0.00	0.00	0.00	0.00
a_{22}	0.00	0.00	0.00	0.00	0.00
a_{23}	0.00	0.00	0.00	0.00	0.00
a_{24}	0.00	0.00	0.00	0.00	0.00
a_{25}	0.00	0.00	0.00	0.00	0.00
a_{26}	0.00	0.00	0.00	0.00	0.00
a_{27}	0.00	0.00	0.00	0.00	0.00
a_{28}	1.63	0.00	1.22	0.00	0.00
a_{29}	0.16	0.00	0.00	0.00	0.00
a_{30}	0.00	0.00	0.00	0.00	0.00
a_{31}	4.53	10.00	4.60	10.00	10.00
a_{32}	6.50	0.00	3.50	0.00	0.00
a_{33}	0.26	0.00	0.00	0.00	0.00
a_{34}	0.00	0.80	0.00	6.00	6.00
a_{35}	0.00	6.25	0.00	0.00	0.00
a_{36}	0.00	5.58	0.00	5.55	2.93
a_{37}	0.00	0.00	0.00	0.00	0.00
a_{38}	0.00	0.00	0.00	0.00	10.00
a_{39}	0.00	0.00	0.00	0.00	10.00
a_{40}	0.00	0.00	0.00	0.00	0.00
a_{41}	0.00	0.00	0.00	0.00	0.00
a_{42}	0.00	0.00	0.00	0.00	0.00
a_{43}	0.00	0.00	10.00	0.00	10.00
a_{44}	0.00	0.00	0.00	0.00	0.00
a_{45}	0.00	0.00	0.00	0.00	0.00
a_{46}	10.00	10.00	10.00	10.00	10.00
a_{47}	0.00	0.00	0.00	0.00	0.00
a_{48}	0.00	0.00	0.00	0.00	0.00
a_{49}	0.00	0.00	0.00	0.00	0.00
a_{50}	0.00	0.00	0.00	0.00	0.00
a_{51}	0.00	0.00	0.25	0.00	0.25

Šios situacijos vaizdžiau pateiktos 4.19 lentelėje.

4.19 lentelė. Realių situacijų pirminių parametrų reikšmingumas

Parametrai	Klasė A	Klasė M		Klasė R	
	S^A_1	S^M_1	S^M_2	S^R_1	S^R_2
a_{1}					
a_{2}					
a_{3}					
a_{4}					
a_{5}					
a_{6}					
a_{7}					
a_{8}					
a_{9}					
a_{10}					
a_{11}					
a_{12}					
a_{13}					
a_{14}					
a_{15}					
a_{16}					
a_{17}					
a_{18}					
a_{19}					
a_{20}					
a_{21}					
a_{22}					
a_{23}					
a_{24}					
a_{25}					
a_{26}					
a_{27}					
a_{28}					
a_{29}					
a_{30}					
a_{31}					
a_{32}					
a_{33}					
a_{34}					
a_{35}					
a_{36}					
a_{37}					
a_{38}					
a_{39}					
a_{40}					
a_{41}					
a_{42}					
a_{43}					
a_{44}					
a_{45}					
a_{46}					
a_{47}					
a_{48}					
a_{49}					
a_{50}					
a_{51}					

4.5.1.1. Realių situacijų atpažinimas pagal pirmojo klasterizavimo rezultatus

Suklasterizavus situacijas, pagal apskaičiuotus pirminius koeficientus buvo išbandytas tų pačių situacijų atpažinimas (4.1.7 skyrius). Tik viena situacija $S_3^{M,3}$ (M-tosios klasės trečiojo panašumo lygmens trečioji situacija) iš visų 343-jų situacijų buvo atpažinta neteisingai. Ji buvo priskirta R-tosios klasės antrajam klasteriui. Trečiajam M-tosios klasės panašumo lygmeniui atstovauja situacija $S_1^{M,3}$ (plačiau apie pirmąjį klasterizavimo metodą 4.2.1 skyriuje). Kadangi po pirmos klasterizavimo iteracijos (apskaičiavus po vieną klasterį kiekvienai klasei), visos M-tosios klasės trečiojo panašumo lygmens situacijos buvo priskirtos savo klasės apskaičiuotam klasteriui, tai antrosios iteracijos metu trečiajam lygmeniui vėl atstovavo situacija $S_1^{M,3}$. Todėl R-tosios klasės antrojo klasterio pirminiai koeficientai gauti nelyginant, kad šios klasės maksimizuojama situacija ir $S_3^{M,3}$ būtų κ-nepanašios. Todėl kai kurie R-tosios klasės reikšmingumo koeficientai išaugo.

Situacijų $S_1^{M,3}$ ir $S_3^{M,3}$ pirminių parametrų įverčiai ir jų reikšmingumas pateikti 4.20 lentelėje.

Didžiausią įtaką, kad situacija $S_3^{M,3}$ buvo neteisingai atpažinta, turi parametras a_{31}. Būtent šio parametro koeficientas R-tosios klasės antrajame klasteryje dominuoja labiau nei M-tosios klasės antrajame klasteryje, kurie buvo apskaičiuoti antrosios iteracijos metu. Bendrai lyginant situacijas $S_1^{M,3}$ ir $S_3^{M,3}$ jos yra γ-panašios, nes lyginant visi parametrai lygiaverčiai ir jų yra pakankamai daug, kad a_{31} parametro įverčiai tose situacijose įtakotų, kad jos nebūtų γ-panašios. Todėl situacijų grupavimas turi tokį trūkumą – vertinamas tik panašumas tarp situacijų, bet ne skirtumas tarp jų.

4.20 lentelė. Situacijų $S_1^{M,3}$ ir $S_3^{M,3}$ pirminiai parametrai ir jų reikšmingumas

Parametrai	Klasė M			Parametrai	Klasė M	
	$S_1^{M,3}$	$S_3^{M,3}$			$S_1^{M,3}$	$S_3^{M,3}$
a_1	0.00	0.00		a_1		
a_2	0.86	0.43		a_2		
a_3	0.00	0.00		a_3		
a_4	0.00	0.00		a_4		
a_5	0.00	0.00		a_5		
a_6	2.50	2.50		a_6		
a_7	3.33	1.67		a_7		
a_8	0.47	0.41		a_8		
a_9	0.01	0.01		a_9		
a_{10}	5.00	5.00		a_{10}		
a_{11}	9.76	9.81		a_{11}		
a_{12}	0.00	0.00		a_{12}		
a_{13}	0.00	0.00		a_{13}		
a_{14}	0.00	0.00		a_{14}		
a_{15}	5.00	2.50		a_{15}		
a_{16}	0.50	0.75		a_{16}		
a_{17}	2.05	2.05		a_{17}		
a_{18}	0.00	0.00		a_{18}		
a_{19}	0.00	0.00		a_{19}		
a_{20}	0.00	0.00		a_{20}		
a_{21}	0.00	0.00		a_{21}		
a_{22}	0.00	0.00		a_{22}		
a_{23}	0.00	0.00		a_{23}		
a_{24}	0.00	0.00		a_{24}		
a_{25}	0.00	0.00		a_{25}		
a_{26}	0.00	0.00		a_{26}		
a_{27}	0.00	0.00		a_{27}		
a_{28}	0.53	0.00		a_{28}		
a_{29}	0.00	0.00		a_{29}		
a_{30}	0.00	0.00		a_{30}		
a_{31}	5.00	10.00		a_{31}		
a_{32}	6.50	0.00		a_{32}		
a_{33}	0.20	0.00		a_{33}		
a_{34}	0.00	0.00		a_{34}		
a_{35}	0.00	0.00		a_{35}		
a_{36}	0.00	2.96		a_{36}		
a_{37}	0.00	0.00		a_{37}		
a_{38}	0.00	5.00		a_{38}		
a_{39}	0.00	0.05		a_{39}		
a_{40}	0.00	0.00		a_{40}		
a_{41}	0.00	0.00		a_{41}		
a_{42}	0.00	0.00		a_{42}		
a_{43}	0.00	0.00		a_{43}		
a_{44}	0.00	0.00		a_{44}		
a_{45}	0.00	0.00		a_{45}		
a_{46}	10.00	10.00		a_{46}		
a_{47}	0.00	0.00		a_{47}		
a_{48}	0.00	0.00		a_{48}		
a_{49}	0.00	0.00		a_{49}		
a_{50}	0.00	0.00		a_{50}		
a_{51}	0.00	0.00		a_{51}		

4.5.2. Realių situacijų koeficientų radimas antruoju klasterizavimo metodu

Skaičiuojant realių duomenų pirminius koeficientus pirmuoju klasterizavimo metodu, klasei A buvo suskaičiuotas vienas klasteris, klasei M – 5, klasei R – 4. Šių klasių visų klasterių koeficientai, maksimizavimo funkcijos ir jų norminimo konstantos parodyti 4.21 lentelėje.

4.21 lentelė. Realių situacijų pirminių koeficientų klasteriai

Pirminiai koeficientai	Klasė A	Klasė M					Klasė R			
	K^A_1	K^M_1	K^M_2	K^M_3	K^M_4	K^M_5	K^R_1	K^R_2	K^R_3	K^R_4
k_1	10.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
k_2	10.00	7.88	3.68	10.00	10.00	10.00	10.00	10.00	0.00	0.00
k_3	0.00	10.00	0.00	0.00	0.00	5.18	10.00	10.00	0.00	2.24
k_4	3.36	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	5.75
k_5	10.00	10.00	0.00	0.00	0.00	4.67	0.00	10.00	0.00	0.54
k_6	0.00	10.00	10.00	0.00	10.00	10.00	8.90	10.00	9.96	0.00
k_7	10.00	0.00	10.00	0.00	10.00	2.84	0.00	10.00	10.00	0.00
k_8	10.00	0.00	10.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00
k_9	0.00	0.00	10.00	10.00	10.00	10.00	0.00	0.00	10.00	0.00
k_{10}	6.57	0.00	3.12	10.00	10.00	1.58	0.00	6.15	3.48	0.00
k_{11}	0.00	0.00	0.00	10.00	10.00	5.64	0.00	10.00	10.00	0.00
k_{12}	10.00	2.25	0.00	0.00	7.72	0.00	1.08	10.00	10.00	1.46
k_{13}	10.00	10.00	0.00	0.00	3.50	10.00	10.00	10.00	10.00	0.53
k_{14}	10.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	3.36
k_{15}	10.00	0.00	10.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00
k_{16}	0.00	10.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00
k_{17}	8.67	0.00	10.00	10.00	10.00	10.00	10.00	6.03	10.00	0.00
k_{18}	4.28	10.00	10.00	7.87	10.00	10.00	0.00	0.00	10.00	1.91
k_{19}	3.16	0.00	6.35	1.98	8.52	0.00	0.00	6.05	5.31	10.00
k_{20}	4.72	0.00	2.27	0.00	0.00	10.00	10.00	7.22	6.06	0.00
k_{21}	3.68	0.00	10.00	10.00	10.00	10.00	10.00	6.12	6.14	2.40
k_{22}	10.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	0.00
k_{23}	1.32	0.00	10.00	10.00	10.00	10.00	3.05	10.00	10.00	0.00
k_{24}	0.00	0.00	10.00	10.00	10.00	10.00	0.00	6.58	8.42	0.00
k_{25}	9.33	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	3.17
k_{26}	0.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	0.00
k_{27}	0.00	0.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00
k_{28}	8.66	0.00	0.00	0.00	7.02	10.00	0.00	0.00	0.00	0.00
k_{29}	0.00	0.00	10.00	0.00	8.41	0.00	4.03	0.00	0.00	3.67
k_{30}	0.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	0.00
k_{31}	0.00	0.98	9.22	4.53	10.00	0.00	10.00	10.00	10.00	10.00
k_{32}	10.00	0.00	0.00	10.00	2.31	0.00	0.00	0.00	0.00	10.00
k_{33}	10.00	0.00	0.00	0.00	8.50	0.00	10.00	0.00	6.54	3.39
k_{34}	0.00	0.00	0.00	9.19	3.04	0.00	10.00	10.00	10.00	3.30
k_{35}	4.69	10.00	10.00	10.00	10.00	2.49	0.00	0.00	0.00	1.66
k_{36}	0.00	10.00	0.00	0.00	10.00	0.00	10.00	0.00	0.00	0.00
k_{37}	6.14	0.00	10.00	8.28	8.01	6.59	10.00	10.00	10.00	0.00
k_{38}	9.01	0.00	4.36	0.00	10.00	2.68	0.00	5.20	3.64	0.00
k_{39}	5.75	0.00	6.96	0.00	10.00	5.14	0.00	4.93	5.94	0.00
k_{40}	0.00	0.00	10.00	10.00	10.00	0.00	10.00	1.65	3.05	0.00
k_{41}	0.00	0.00	10.00	10.00	10.00	0.00	10.00	10.00	10.00	0.00
k_{42}	0.00	0.00	1.67	6.53	10.00	0.00	10.00	10.00	10.00	0.00
k_{43}	1.17	0.00	1.71	0.46	7.92	10.00	1.99	6.88	5.47	3.79
k_{44}	9.92	0.00	0.00	0.00	10.00	0.00	8.32	10.00	10.00	1.72
k_{45}	0.00	0.00	0.00	0.00	10.00	0.00	10.00	4.42	10.00	0.00
k_{46}	3.88	0.00	10.00	10.00	10.00	10.00	10.00	6.35	10.00	0.00
k_{47}	0.00	0.00	0.00	0.00	10.00	0.00	10.00	0.00	7.32	0.00
k_{48}	3.93	0.00	0.00	0.00	10.00	0.00	10.00	6.00	6.25	2.11
k_{49}	4.71	0.00	0.00	0.00	10.00	0.00	10.00	5.81	5.99	1.64
k_{50}	0.00	0.00	0.00	0.00	0.56	0.00	0.00	2.63	3.77	0.00
k_{51}	2.79	0.00	5.76	4.74	8.38	10.00	10.00	6.31	6.51	0.28
Max funkcijos	Φ^A_{1max}	Φ^M_{1max}	Φ^M_{2max}	Φ^M_{3max}	Φ^M_{4max}	Φ^M_{5max}	Φ^R_{1max}	Φ^R_{2max}	Φ^R_{3max}	Φ^R_{4max}
	171.16	245.25	113.93	136.56	79.26	135.68	157.28	169.83	115.57	156.07
Konstantos	c^A_1	c^M_1	c^M_2	c^M_3	c^M_4	c^M_5	c^R_1	c^R_2	c^R_3	c^R_4
	1.43	1.00	2.15	1.80	3.09	1.81	1.56	1.44	2.12	1.57

Vaizdžiau pirminiai koeficientai pateikti 4.22 lentelėje.

4.22 lentelė. Realių situacijų pirminių koeficientų reikšmingumas

Pirminiai koeficientai	Klasė A	Klasė M					Klasė R			
	K^A_1	K^M_1	K^M_2	K^M_3	K^M_4	K^M_5	K^R_1	K^R_2	K^R_3	K^R_4
k_1										
k_2										
k_3										
k_4										
k_5										
k_6										
k_7										
k_8										
k_9										
k_{10}										
k_{11}										
k_{12}										
k_{13}										
k_{14}										
k_{15}										
k_{16}										
k_{17}										
k_{18}										
k_{19}										
k_{20}										
k_{21}										
k_{22}										
k_{23}										
k_{24}										
k_{25}										
k_{26}										
k_{27}										
k_{28}										
k_{29}										
k_{30}										
k_{31}										
k_{32}										
k_{33}										
k_{34}										
k_{35}										
k_{36}										
k_{37}										
k_{38}										
k_{39}										
k_{40}										
k_{41}										
k_{42}										
k_{43}										
k_{44}										
k_{45}										
k_{46}										
k_{47}										
k_{48}										
k_{49}										
k_{50}										
k_{51}										

4.5.2.1. Realių situacijų atpažinimas pagal antrojo klasterizavimo rezultatus

Suklasterizavus situacijas, pagal apskaičiuotus pirminius koeficientus buvo išbandytas tų pačių situacijų atpažinimas (4.1.7 skyrius). Kaip ir atpažįstant situacijas pagal koeficientus, gautus skaičiuojant pirmuoju klasterizavimo metodu, tik viena situacija buvo neteisingai atpažinta. Situacija $S_6^{R,6}$ (R-tosios klasės šeštojo panašumo lygmens šeštoji situacija) buvo priskirta M-tosios klasės ketvirtajam klasteriui. Šeštajam R-tosios klasės panašumo lygmeniui atstovauja situacija $S_1^{R,6}$. M-tajai klasei koeficientai skaičiuojami lyginant, kad šios klasės maksimizuojamoji situacija su $S_1^{R,6}$ būtų κ-nepanašios, bet nelyginama, kad maksimizuojamoji situacija su $S_6^{R,6}$ būtų κ-nepanašios (plačiau apie antrąjį klasterizavimo metodą 4.2.2 skyriuje). Todėl kai kurie M-tosios klasės klasterių koeficientai išaugo.

Situacijų $S_1^{R,6}$ ir $S_6^{R,6}$ pirminių parametrų įverčiai ir jų reikšmingumas pateikti 4.23 lentelėje.

Didžiausią įtaką, kad situacija $S_6^{R,6}$ buvo neteisingai atpažinta, turi parametrai a_{36} ir a_{39}. Būtent šių parametrų koeficientai labiausiai dominuoja M-tosios klasės ketvirtajame klasteryje. Situacija $S_6^{R,6}$, kaip ir atpažinime pagal pirmojo klasterizavimo rezultatus (skyrius), buvo neteisingai atpažinta, nes situacijų grupavimas vertina tik panašumą tarp situacijų, bet ne skirtumą tarp jų.

4.23 lentelė. Situacijų $S_1^{R,6}$ ir $S_6^{R,6}$ pirminiai parametrai ir jų reikšmingumas

Parametrai	Klasė M			Parametrai	Klasė M	
	$S_1^{R,6}$	$S_6^{R,6}$			$S_1^{R,6}$	$S_6^{R,6}$
a_1	0.00	0.00		a_1		
a_2	1.00	1.00		a_2		
a_3	0.00	0.00		a_3		
a_4	0.00	0.00		a_4		
a_5	0.33	0.00		a_5		
a_6	2.50	2.50		a_6		
a_7	1.67	1.67		a_7		
a_8	0.78	0.87		a_8		
a_9	0.00	0.00		a_9		
a_{10}	0.00	0.00		a_{10}		
a_{11}	0.53	0.15		a_{11}		
a_{12}	0.00	0.00		a_{12}		
a_{13}	2.50	2.50		a_{13}		
a_{14}	0.00	0.00		a_{14}		
a_{15}	2.50	2.50		a_{15}		
a_{16}	0.00	0.00		a_{16}		
a_{17}	2.05	2.05		a_{17}		
a_{18}	0.00	0.00		a_{18}		
a_{19}	1.00	1.00		a_{19}		
a_{20}	0.00	0.00		a_{20}		
a_{21}	0.00	0.00		a_{21}		
a_{22}	0.00	0.00		a_{22}		
a_{23}	0.00	0.00		a_{23}		
a_{24}	0.00	0.00		a_{24}		
a_{25}	0.00	0.00		a_{25}		
a_{26}	0.00	0.00		a_{26}		
a_{27}	0.00	0.00		a_{27}		
a_{28}	0.00	0.00		a_{28}		
a_{29}	0.00	0.00		a_{29}		
a_{30}	0.00	0.00		a_{30}		
a_{31}	10.00	10.00		a_{31}		
a_{32}	0.00	0.00		a_{32}		
a_{33}	0.00	0.00		a_{33}		
a_{34}	6.00	6.40		a_{34}		
a_{35}	0.00	0.00		a_{35}		
a_{36}	2.93	6.23		a_{36}		
a_{37}	7.50	0.00		a_{37}		
a_{38}	10.00	10.00		a_{38}		
a_{39}	0.50	10.00		a_{39}		
a_{40}	0.00	0.00		a_{40}		
a_{41}	0.00	0.00		a_{41}		
a_{42}	0.00	0.00		a_{42}		
a_{43}	10.00	10.00		a_{43}		
a_{44}	0.00	0.00		a_{44}		
a_{45}	0.00	0.00		a_{45}		
a_{46}	10.00	10.00		a_{46}		
a_{47}	0.00	0.00		a_{47}		
a_{48}	0.00	0.00		a_{48}		
a_{49}	0.00	0.00		a_{49}		
a_{50}	0.00	0.00		a_{50}		
a_{51}	0.25	0.25		a_{51}		

5. IŠVADOS

Atlikus tyrimus: a) su dirbtiniais duomenimis, b) su konkrečiais realiais kontraktais, ir pritaikius darbo kūrimo technologiją, galima padaryti tokias išvadas:

1) eksperimentinis tyrimas rodo, kad galima sukurti miglotai apibrėžtos situacijos įvertinimo modulį, kurį būtų lengva integruoti bet kurioje su kreditavimo procesais susijusioje kompanijoje, ir kuris iš kontraktų apdorojimo istorijos (ypač iš pasibaigusių kontraktų) apskaičiuotų parametrų koeficientus, įvertindamas ir jų tarpusavio sąryšius, ir panašumus bei skirtumus tarp sėkmingų ir nesėkmingų kontraktų;
2) atliktas situacijų norminimas leidžia pasiekti, kad situacijų savybių įverčiai būtų tarpusavyje išmatuojami ir palyginami;
3) atliktas situacijų grupavimas pagal jų tarpusavio panašumą tam, kad sumažinti situacijų kiekį klasėje. Tačiau paprastai pats situacijų grupavimas turi trūkumą – vertinamas tik panašumas tarp tos pačios klasės situacijų, bet ne skirtumas tarp jų;
4) sudaryti ir realizuoti taikymo sričiai (patyriminės istorijos kontraktams) tinkami du klasterizavimo metodai;
5) realizuotas tiesinio programavimo uždavinio formulavimo metodas, atitinkantis taikymo sritį ir įvertinantis panašumus tarp tos pačios klasės situacijų bei skirtumus tarp skirtingų klasių situacijų;
6) realizuotas atpažinimo algoritmas (situacijų priskyrimas panašiausiai klasei), naudojantis situacijų savybių reikšmingumo koeficientus;
7) sudaryta miglotai apibrėžtos situacijos įvertinimo modulio funkcinė organizacija.

6. LITERATŪRA

[1] CHIN-TENG LIN; C.S. GEORGE LEE *Neural fuzzy systems.* Prentice-Hall, Inc, 1996.

[2] ZADEH, L.A. *Fyzzy Sets.* Information and Control, vol. 8. p. 338-353. 1965.

[3] LANDAU, L. *Fuzzy Thinking* [interaktyvus] 2004, birželis [žiūrėta 2006-04-14]. Prieiga per internetą: http://www.mth.kcl.ac.uk/events/summer_schools/summer_school2004/Landau.html

[4] SHORTLIFFE, E. H. *Computer-Based Medical Consultation*: MYCIN. New York: Elsevier North-Holand. 1976.

[5] KEVIN M. PASSINO; STEPHEN YURKOVICH *Fuzzy Control.* Addison Wesley Longman, Inc, 1998.

[6] ZADEH, L. A. *The concepts of a linguistic variable and its application to approximate reasoning, Part 1.* Inf. Sci. 8:199-249. 1975.

[7] BEZDEK, J. C. AND S.K. PAL, eds. *Fuzzy Models for Pattern Recognition.* New York: IEEE Press. 1992.

[8] BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press. 1981.

[9]ЯСИНЯВИЧЮС, Р. Ю. *Параллельные пространственно-временные вычислительные структуры*. Вильнюс. 1988.

[10] MERKEVIČIUS, E.; GARŠVA, G. *Kreditavimo taisyklių reikšmingumo įvertinimas savitvarkių žemėlapių metodu*. 2005.

[11] MERKEVIČIUS, E.; GARŠVA, G.; SIMUTIS, R. *Forecasting of Credit Classes With the Self-Organizing Maps*. Informacinės technologijos ir valdymas. 2004. ISSN 1392-124X.