әл-Фараби атындағы Қазақ ұлттық университеті

ӘОЖ 004.93ʼ1 Қолжазба құқығында

ЧЕРИКБАЕВА ЛЯЙЛЯ ШАРИПОВНА

Топтық шешімдердің тиімді алгоритмдерін тану есептерінде зерттеу және өңдеу

6D070300 – Ақпараттық жүйелер

Философия докторы (PhD)
дәрежесін алу үшін дайындаған диссертация

Ғылыми кеңесшілер:
ҚР ҰИА және ХАА академигі,
техника ғылымдарының докторы,
профессор Амиргалиев Е.Н.,
РҒАСБ математика институты
техника ғылымдарының докторы,
профессор Бериков В.Б.

Қазақстан Республикасы
Алматы, 2020

МАЗМҰНЫ

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР

АЖ – ақпараттық жүйе
ЖК – жақын көрші
DM – деректер қоры
SVM – тірек векторларының әдісі
RGB – спектрлердің қызыл, жасыл, көк түсі
БК – Бақылау комитеті
РҒА ЕО – Ресей Ғылым Академиясы Есептеу Орталығы
TN – True Negative, бинарлы классификациялау есебінде дұрыс анықталған теріс кластардың объектілер саны
TP – True Positive, бинарлы классификациялау есебінде дұрыс анықталған оң кластардың объектілер саны
ГСБ – гиперспектральды бейне
IT – Ақпараттық технологиялар
ЖҚЗ – Жерді қашықтықтан зондтау
ҒК – ғылым комитеті

КІРІСПЕ

Елімізде 2017 жылдың қаңтар айының 31-де Елбасының «Қазақстанның үшінші жаңғыруы: жаһандық бәсекеге қабілеттілік» деп аталатын Қазақстан халқына Жолдауы жарияланды. Осы жолдау негізінде 2017 жылғы 12 желтоқсанда Қазақстан Республикасы Үкіметінің №827 Қаулысымен «Цифрлық Қазақстан» мемлекеттік бағдарламасы бекітілді. Жолдауда еліміздің ұзақ мерзімді мақсаттарды және аз уақыт ішінде шешілуі қажет міндеттер нақты анықталды. Сол міндеттердің ішіндегі елімізде IT-саласын дамыту мәселесі туралы: «Біз цифрлық технологияларға негізделген болашағы зор бағыттарға айрықша мән беретін боламыз» - деп, IT-саласын дамытуды ерекше бақылауда ұстауымыз қажеттілігін атап айтты. Атап айтқанда медицина [1], қаржы, білім және т.б салаларда кеңінен қолданыс тауып, халықтың әлеуметтік жағдайын жақсартуға септігін тигізуде. Цифрландыру процесін жүзеге асыру еліміздің бәсекеге қабілетті елдер қатарына енуіне мүмкіндік береді. Бұл процесті жүзеге асыру мақсатында бес бағыт бойынша жұмыс жасау қабылданды: экономикалық салаларды сандық түрге келтіру, сандық мемлекетке көшу, сандық жібек жолын іске асыру, адами капиталды дамыту, инновациялық экожүйелерді құру [2]. Бұл бағыттардағы сандық деректердің үлкен көлемі және олардың құрылымданбауы ақпараттық кедергілер, мәселелерге тудырады. Сондықтан да осы мәселелерді шешуде жаңа шешімдерді, жаңа тәсілдерді, сонымен қатар ақпараттарды өңдеуші жаңа алгоритмдер қалыптастыру қажет.

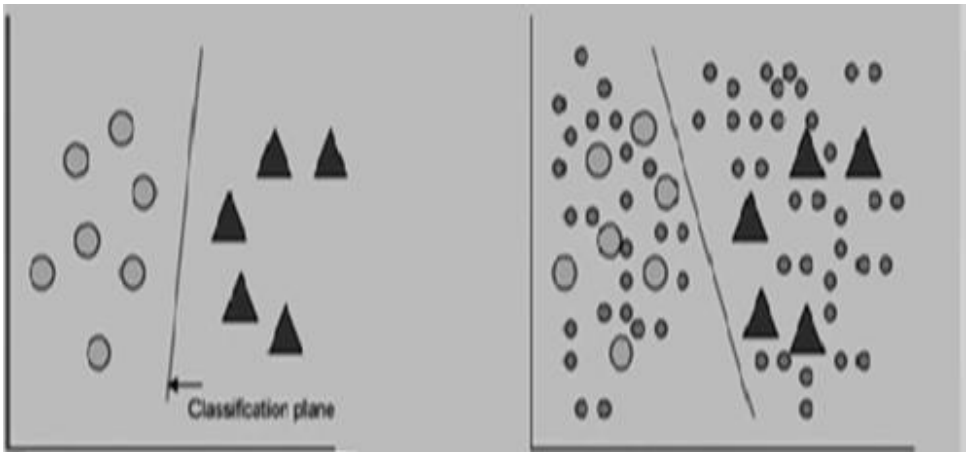
Тақырыптың өзектілігі. Бұл күнде шешімдерді қабылдау саласындағы ақпараттық технологиялардың дамуы сауықтыру, байланыс, сараптамалық талдау жасау, өндіріс, ғаламтор және т.б. салалардың ажырамас бөлігі болып табылатын үлкен көлемді, көп өлшемді және күрделі құрылымды деректерді өңдеуге бағытталған деректерді (Data Mining, DM) интеллектуалды талдаудың программалық құрылғыларын жасауға қатысты.

Деректерді интеллектуалды талдау жүйесінде классификациялау және бейнені тану мәселесі ерекше орын алады, өйткені объектілерді бөлуді жүргізу қажеттілігі медициналық диагностикада, кредит төлеу қабілеттілігін анықтауда, қолжазбалық символдарды тануда, текстерді категорияларға бөлуде, ақпараттарды алуда және т.б. салаларының қолданбалы есептерін шешуде жиі кездеседі. Сонымен қатар Жерді қашықтықтан бақылаумен алынған суретті өңдеу, соның ішінде гиперспектральды суретті өңдеу, суретке түсірілген әртүрлі объектілерді (адам беті, жаяу жүргіншілер өткеліндегі және т.б.) идентификациялауда және нақты уақыт режиміндегі видеоларды талдауда жүргізілген классификациялаудың да актуалдылығы кем емес.

Бейне тану және классификациялау есептері өздеріңіз білетіндей, ертеректе, осыдан 30-40 жыл бұрын ғылымның әр саласында жеке-жеке алгоритмдері арқылы көптеген мәселелер шешіліп отырған. Бұл алгоритмдер нақты бір есепті шешуге арналғандықтан жалпы немесе басқа типті есептерді шешу барысында көптеген бір кемшіліктерге ие болуы мүмкін. Сондықтан да қазіргі уақытта бейне тану мен классификациялау алгоритмдерінің дамуының

келесі сатысы топтық шешімдер алгоритмдері. Топтық шешімдер алгоритмдерін қолдануға тағы бір негіз болатын себептері: алгоритмдер жұмысының нәтижелерінің тұрақтылығы артады; топтаудың нақты бір қойылған шартын орындауда алынған нәтижелер сапасы артады; алгоритмдердің параметрлерін таңдаудан тәуелділік төмендейді; алынған шешімдердің сапасы жақсара түседі.

Классификациялау және тану есебін қоюдың бірнеше нұсқасы бар: бақыланатын оқыту, бақыланбайтын және жартылай бақыланатын оқыту және оларға мысал 1- суретте келтірілген.



Сурет 1 – Бақыланатын және жартылай бақыланатын оқыту мысалы

Бақыланатын оқытуда берілген объектілер үшін кластар белгілі болады. Сәйкесінше бақыланбайтын оқытуда кластары белгісіз болады. Ал жартылай бақыланатын оқытуда берілген объектілер үшін кластар жартылай белгілі.

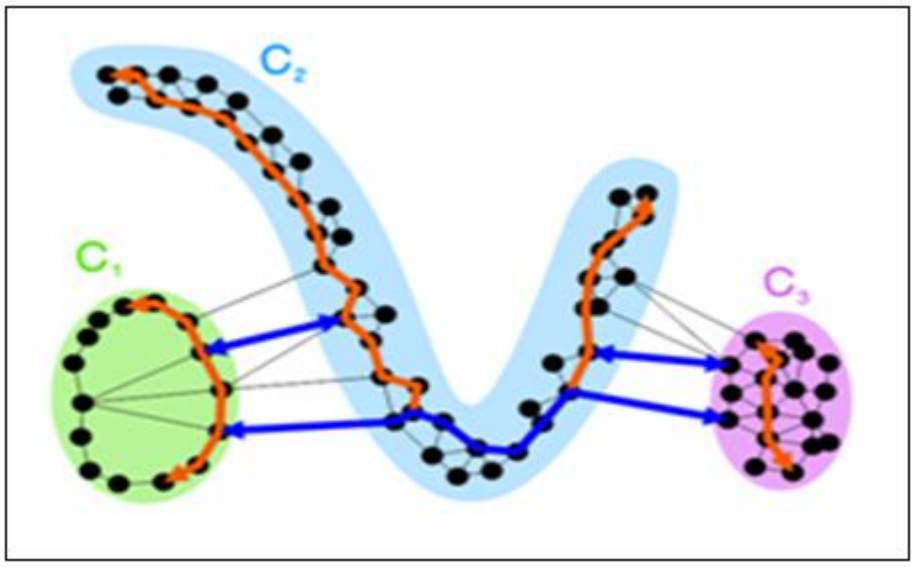
Берілген жұмыста бейнені тану есебін қоюдың бір нұсқасы – жартылай бақыланатын оқыту жағдайындағы классификациялау есебі (*semi-supervised classification*) қарастырылады. Бұл есепте берілген объектілерінің бір бөлігі үшін кластар белгілі; мұнда қолда бар белгіленбеген объектілерді классификациялау керек немесе жаңа объектілерді тану үшін шешуші ережені қалыптастыру керек. Берілген есептің өзекті болуының тағы бір себептері:

- белгіленбеген деректер «шығын аз болады»;
- белгіленбеген деректерді белгіленген деректермен бірге қолдану машиналық оқыту сапасының айтарлықтай өсуін қамтамасыз етеді.

Неліктен топтық шешімдер алгоритмі қажет болып отыр, неге біз бір ғана алгоритмдермен есептерді шеше бермейміз? Себебі есептер түрлері өзгеріп жатыр, сонымен қатар деректерде белгісіздер, анықталмағандықтар көбейіп отыр. Өмірдегі экологиялық, технологиялық, экономикалық процестерде, геологияда, метрологияда, денсаулық сақтау салаларында бейне тану алгоритмдері, соның ішінде топтық шешімдер алгоритмдерін қолдану жақсы нәтиже береді деп ойлаймыз. Бұл мәселе өте үзіліссіз жетілдірілуде, қарастырылуда. Диссертациялық жұмыста осы мәселелерді шешуде біз әдістер мен екі алгоритмді ұсынып, программалық қамтама дайындап,

сонымен қатар барлығын ақпараттық жүйе аясында топтастыра отырып, жүйе құрастырдық.

Атап айта кету керек, әмбебап кластерлік талдау, классификациялау әдістері және алгоритмдері жоқ. Сонымен қатар әртүрлі алгоритмдерді бір объектілер жиынына қолдануда әртүрлі нәтижелер аламыз. Өйткені әр алгоритмнің негізінде моделдеудің әртүрлі принциптері бар, сонымен қатар қолданылып отырған метрика, жақындық функциялары, тиімділік критерилері, бастапқы параметрлердің таңдалынуы, әртүрлі типті сипаттамалармен жұмыс жасау әдістеріне байланысты болады. Сондықтан да әрбір жеке алгоритмдерден қарағанда қателік саны аз, бірнеше алгоритмдердің топтау нәтижесінде алынған, объектілерді бөлу нәтижелерін біріктіретін нәтижелік классификациялау шешімін алу қажеттілігі туындайды.



Сурет 2 – Әртүрлі деректер құрылымы

Берілген 2-суреттен көріп отырғандарыңыздай, деректер құрылымы әртүрлі, көлемі үлкен заманауи деректер қорында (Big Data) бірқатар объектілер шар пішінде шоғырлана орналасатын болса, кейбіреулері спираль түрінде, үлестірілген түрде орналасулары мүмкін. Көлемі өте үлкен заманауи деректер қоры (Big Data) және оларға арнайы өңдеу әдістері қажет. Қолданыста бар классификаторларды қолдану бірқатар күрделіліктерді туындатады. Біріншіден, бұл классификаторлар бір-біріне ұқсас емес, екіншіден, қолданбалы зерттеу бағыттарын өзгерту динамикасы аса жоғары, сонымен қатар классификаторлар қажеттілігінше тез өзгеріп үлгермейді [3].

Тану есебі – қарастырылып отырған объектілер, құбылыстарды белгілері бойынша қажетті топтарға бөлу. Әр объект белгілердің жиынымен сипатталады. Классификациялау жататын кластары белгілі прецценттер – объектілер негізінде жүргізіледі. Ақпараттарды өңдеу мен тарату өткен ғасырдың ортасында өңделуші ақпараттарды машиналарға тану мүмкіндігін беретін технологияларға деген қажеттілік арта түсті. Оларға текстерді тану, машиналық көру, сөйлеуді тану, медицинада ауруды анықтау, саусақ ізін

тануды мысал ретінде алуға болады. Осы есептердің кейбіреулерін жоғары жылдамдықта айтарлықтай деңгейде адамдар шеше алғанымен де, осы уақытқа дейін оларды жалпы түрде шешуші компьютерлік программалар жасалынбаған. Осыған байланысты, бейнені тану мәселесі барлық жерде, соның ішінде жасанды интеллект және роботтық техника саласында тарала бастады [4].

Тану есептері мен классификациялау теориясы аумағындағы прецденттер бойынша алғашқы жұмыстар өткен ғасырдың басында жарық көре бастады және ол классификациялау есебіне бөлуші функциясын қолданумен жасалынған шешім қабылдаудың байестік теориясымен (Э.Пирсон, Дж.Нейман жұмыстары), гипотезаны тексеру мәселелерін шешумен байланысты жұмысы (А. Вальд) болды. КСРО-да бейнені тану саласындағы алғашқы жұмысты 1959 ж. заманауи ақпараттар теориясының негізін салушылардың бірі А.А. Харкев жасады. Бейнені танудың теориясы мен практикасын дамытуға Л. Заде, Я.З. Цыпкин, А.Г. Ивахненко (аргументтерді топтық есепке алу әдісі), Ю.И. Журавлев, В.А. Ковалевский, Н.Г. Загоруйко (таксономия және білімдер талдауының алгоритмдері), Э.М. Браверман, Л.И. Розоноэр (потенциалдық функциялар әдісі), М.М. Бонгард, А.Я. Червоненкис (танудың статистикалық теориясы), В.Д. Мазуров (комитеттер әдісі), Г.С. Лбов (тәуелділікті іздеу және танудың логикалық әдістері) айтарлықтай өз үлестерін қосты. Қарқынды зерттеу жұмысы КСРО ЕО ҒА (ВЦ АН СССР)-да XX-ғасырдың 60-жылдарының соңынан бастау алды. Осы жылдарда академик Ю.И. Журавлев оқытушы прецденттер аз болған жағдайында тану есебін тиімді шешуші тану алгоритмін ұсынды. Ары қарай осы алгоритм негізінде Ю.И. Журавлев танушы процедуралардың жаңа класы – бағаларды есептеу алгоритмін құрды, содан кейін танудың алгебралық теориясы зерттеліпіп, енгізілді. Кластерлік талдау аумағындағы бірінші ғылыми топ Н.Г. Загоруйконың басшылығымен Новосибирск қаласында құрылды [5].

АҚШ-да бейнені тану саласының негізін қалаушы, перцептронды – бейнені танумен байланысты ми қызметінің моделін құрушы Розенблатт болды. Бейнені тану саласындағы алғашқы жұмыс сонымен қатар перцептрондарды зерттеу негізінен оқитын автоматтарды құрудың теориясы мен практикасына арналды [6].

Осы уақытқа дейін күрделі құрылымды әртүрлі деректермен жұмыс істеуші көптеген алгоритмдер бар. Олардың арасында осы күнде аса танымал және кең қолданысқа ие болған *k*-means алгоритмін 1956-57 жылдары математиктер Г. Штейнгауз және С. Лойд біруақытта жасады. Танудың барлық алгоритмдерін эвристикалық деп атауға болады.

Соңғы жылдары көптеген классификациялық есептерді шешуде прецеденттер бойынша оқытуды жүзеге асырушы және шекаралық алгоритмдер тобы мен классификациялау тәсілдеріне жататын тірек векторларының алгоритмі қолданылуда. Бұл алгоритм 1963 жылы кездейсоқ сызықты классификациялау үшін ұсынылды. Уақыт өте өткен ғасырдың

соңында Вапник және шәкірттері сызықты емес бөлгішті ойлап тапты. Мұнда негізгі идея скаляр көбейтіндіден ядроға ауысу болып табылады.

Қазіргі уақытта бейнені тану теориясы информатиканың қарқынды түрде дамушы тармағы болып табылады. Бұл жерде қолданылатын әдістер қолданбалы математиканың әртүрлі бөлімдерімен байланыстырады, ал өте ауқымды түсінікке ие «класс» ұғымы оны қолданудың басқа да жаңа аумағын табуға мүмкіндік береді.

Заманауи кластеризациялау теориясының негізгі принципі шет елдік авторлардың жұмыстарына негізделеді: Дж. Рубин, С. МакНотон, Д. Дюран, П. Оделла, Х. Фридман және т.б.

Беждека Дж.К., Данна Дж.К., Дейва Р.Н., Келлер Дж.М. жұмыстарында анықталмағандық шартындағы кластерлеу деп аталатын алгоритмдері (нақты емес с-орташа алгоритмі және оның модификациясы) ұсынылған. Батистакис Я., Бени Г., Галда Х. сияқты шет елдік авторлардың зерттеу жұмыстары алгоритмдер деректерін қолдана отырып кластеризациялау сапасының көрсеткіштерін зерттеуге және құруға арналған.

Бүгінгі таңда топтық шешімдер алгоритмдерімен айналысушы ғалымдар: Ю.И. Журавлев, В.В. Рязанов, Г.С. Лбов, А.С. Бирюков (Москва қ), В.Д. Мазуров (Свердловск қ.), А.Г. Ивахненко (Киев қ.), М.Н. Қалимолдаев, М.Б. Айдарханов, И.А. Мухамедғалиев, А.Е. Дюсембаев, Е.Н. Амиргалиев (Алматы қ.), Carlos Guestrin (США) және басқалар. Жартылай бақыланатын оқыту есебімен айналысушы ғалымдар: Н.Г. Загоруйко, И.А. Пестунов, В.Б. Бериков (Новосибирский қ.), Yu.G., Yao G.(ҚХР) және басқалар.

Барлық бағыттарда тану есебі мен кластерлік талдауды ұқсас деп қарастыруға болады. Әдетте бейне тану теориясы классификациялау теориясымен қамтылады, яғни көп жағдайда классификациялау есебі тану есебі болып алынады(қарастырылады), немесе керісінше. Мысалы, «Жақын көрші» алгоритмі классификациялау есебінде шешуші ереже құру үшін қолданылады.

Ірі және орташа кәсіпорындарда, сонымен қатар ғылыми салаларда пайда болатын ақпараттық процестердің жұмысы кезінде алынған, сақталынған және өңделінген ақпараттардың өсуімен оларды алынған түрінде өңдеу қиындықтар туғызады. Сипаттамалық белгілерін ерекшелеу, құрылымдау, жалпылау және сұрыптау үшін ақпараттарды бастапқы өңдеу қажеттілігі туындайды. Сондықтан ақпараттарға мамандармен кезекті талдау жүргізуге толықтай қажетті өңдеу жүргізуге мүмкіндік беретін классификациялау және кластерлік талдау процестері жүргізіледі [7-8].

Бүгінгі таңда кластеризациялау деректерді өңдеудің бірінші қадамы болып табылады. Қазіргі уақытта кластарға бөлудің көптеген алгоритмдері және олардың модификациялары бар. Бірақ бұл алгоритмдерді бір объектілер жиынына қолдануда әртүрлі шешім аламыз. Кластерлеу есебі шешімінің бірмәнді болмауының себептері төменде көрсетілген:

1. Кластерлер саны көп жағдайда алдын-ала белгілі бола бермейді, оларды көп жағдайларда алгоритмдер талдау процесін бастау кезінде орнатады.

2. Кластерлеу қорытындысы таңдап алынған метрикаларға өте тығыз байланысты болып келеді.

3. Кластерлік талдаудың актуалды мәселелеріне оның шешімдерінің орнықтылығын жоғарылату жатады, алгоритмдер нәтижелері бастапқы қойылатын шарттарды, параметрлерді таңдауға байланысты өзгеріп отырады.

Кластерлік талдаудың келесі негізгі этаптарын атап өтуге болады:

Белгілер жүйесін қалыптастыру. Берілген оъектілер жиынына классификациялау жасауда мамандар белгілердің қайсысымен талдау жасау керектігін нақты анықтай алмайды. Белгілердің бастапқы тобынан тиімді кішігірім, ішкі жүйені таңдау алудың қажеттілігі бола бермейді..

Өткен ғасырдың 70-жылдардың басында Ресей Ғылым Академиясының академигі Ю.И.Журавлев ұсынған, бағаларды есептеу алгоритмі деп аталатын, танудың алгоритмдер класы көмегімен сапа функционалын тиімді таңдауға және ақпараттық белгілерді таңдап алу мақсатында белгілер кеңістігін зерттеуге мүмкіндік береді. Бағаларды есептеу алгоритмдері класын сипаттау үшін бастапқы ақпараттар түсінігін, бағаларды есептеу моделін, сәйкестендірілген объектілер мен кластар жақындығын бағалауды есептеу ережесін анықтау керек. Төменде бағаларды есептеу алгоритмдерінің моделінде қолданылған кейбір қағидалар келтірілген:

- Объектіні классификациялау туралы шешім объектілердің кластарға жақындығын бағалауды талдау көмегімен қабылданады. Қай класқа жақындық бағасы жоғары болса – объект сол класқа жатады. Бағаны танушы оператор есептейді.
- Кластарға жақындық бағасын есептеуде барлық объектілердің эталондық объектілерге қаншалықты жақындығы/алыстығы ескеріледі. Жақындық – сипаттамалардың ұқсастығы, белгілер мәндерінің арасындағы аз ара қашықтық. Сондықтан класқа объектінің жақындық бағасы жоғары болса, ол берілген кластың эталондық объектісіне жақын және басқа кластардың эталондық объектілерінен алыс.
- Бағаларды есептеу алгоритмін анықтауда объектілердің ұқсастықтарын талдау жүргізілетін тірек жиындарының жүйесі құрылады, объектілер жиындарының арасында жақындық ара қашықтықтары есептелінеді. Объектілер жұбы үшін алынған бағалардан белгіленген белгілер жиындары бойынша кластарының бағалар шамасы алынады да кластардан әр кластар үшін қосынды бағасы құрылады. Осылайша, жақындық функциясы, бағаларды есептеу ережесі және шешуші ереже түріндегі тірек жиындарының жүйесін таңдау тәсілі бағаларды есептеу алгоритмдерінің ішкі класын анықтайды, ал сәйкес параметрлер мәнін беру нақты алгоритмді береді. Бағаларды есептеу алгоритмі түріндегі алгоритмдерде объектілердің ұқсастық дәрежесі жеке белгілердің тізбектей сәйкестендірілуіне емес, объектілер сипаттамасына кіретін белгілердің барлық мүмкін болатын белгілерінің сәйкестіктерін салыстыру болып табылады. Бұл жерде атап айта кету керек, бейнені тану кезінде объектілерді қандай белгісі немесе қандай комбинациясы бойынша сәйкестендіру керек. Бұл, көп жағдайда, тану есебін шешу кезінде нәтиженің жоғары болуына әсері мол.

Объектілер немесе объектілер топтары арасындағы қашықтықты есептеу тәсілін анықтау.

Бұл тәсіл қарастырылып отырған шешілуші қолданбалы есептің спецификасын анықтап беру керек.

Объектілерді топтау. Топтау алгоритмдерінің көптеген түрлері бар. Осы алгоритмдерге негізделген негізгі тәсілдер:

- Ықтималдық тәсіл: Берілген объектілердің кездейсоқ класқа жатады деп берілсін, өйткені класс номері белгісіз. Әр класс үшін берілген жиынтықтың ықтималды түрде үлестірілуі анықталған; үлестірілу параметрі белгісіз.
- Графтар теориясына негізделген тәсіл: Бұл жиынтықтың танымал алгоритмі – ең қысқа тұйық емес жол алгоритмі [9]. Алдын-ала графтың объектілері ұштарымен, ал қабырғалары сәйкес объектілер арасындағы тең арақашықтыққа тең ұзындыққа сәйкес келетіндей негізгі ағашы құрылады. Құрылған ағаштан кластерлерді құру үшін ең ұзын қабырғасы алынып тастап отырады.
- Иерархиялық тәсіл: Берілген бағыт теоретикалық-графтық тәсілмен байланысты. Топтау нәтижелері топтаулар ағашы түрінде (дендограммалар) түрінде көрсетіледі. Осы тәсілге негізделген алгоритмдерді агломеративті (жақын топтарды немесе объектілерді этап бойынша біріктіріп отырушы) және дивизмді (бастапқы топты алыстатылған топтарға бөлуді этап бойынша жүргізіп отыру; олар өз кезегінде ары қарай ішкі топтарға бөліне береді және т.с). Топтық шешім кірістірілген иерархиялық ішкі топтар сияқты. Иерархиялық алгоритм нәтижесінде дендрограмма құрылады, яғни, бұтақтары кластер болатын арнайы ағаш. Сіздің деректеріңіз үлкен болса және оның құрылымы бізге белгісіз болған жағдайда иерархиялық кластерді пайдаланған ыңғайлы. Иерархиялық емес алгоритмдерде, иерархиялықтардан ерекшелігі, кластерлердің саны туралы болжамалы түрде гипотеза болу керек және оларды көрсету керек.

Нәтижелерді айқындау. Нәтижелік кластерлер қарапайым, түсінікті, ақпаратты болуы керек. Мұнда кластер деп сәйкес топтар нүктелерінен және кейбір берілген формалардан тұратын минималды көлемді айнымалылар кеңістігінің ішкі аймағын айтады.

Алынған топтаудың сапасын анықтау. Қолданбалы аймақ маманына құрылған топтар зерттелуші объектілердің жаңа қасиеттерін анықтауға көмектеседі ме, талдау мақсатына жетуге мүмкіндік бере ме, ішкі заңдылықтарды көрсете алады ма екендіктеріне көз жеткізу керек. Сапасын тексерудің басқа да нақты үлестіру үлгісінде есептелуі мүмкін топтардың кездейсоқ құрылу ықтималдығын табумен (әртүрлі кластардың біртектілігін бақылау туралы статистикалық гипотезаларын тексеру арқылы) байланысты; бутстрэп әдіспен; әртүрлі сапа көрсеткіштерін есептеумен (ішкі топтық шашырау, С-индекс, Гудман-Крускаль индексі; Ранда индексі және т.б.) байланысты формальды әдістері көп [10].

Кластерлік талдау жүргізудің негізгі мақсаты басқа кластардан барынша өзгеше және топ ішінде өзара ұқсас объектілер тобының қажетті санын бөліп алу. Мұндай талдау деректердегі заңдылықтарды табу үшін ақпараттық

жүйелерде кеңінен қолданылады. Бейнені тану мен классификациялау саласы дамыған 1980-шы жылдары классификаторлар жиынын қолданудың қосымша артықшылықтары пайда бола бастады. Осы уақытқа дейін бейне тану мен классификациялаудың көптеген әртүрлі әдістері тәжірибе жүзінде тексеріліп, жасалынған еді: автоматтар теориясы терминдерінде модельдерді қалыптастырудан тұратын статистикалық әдістер; нейрожелілік тәсілдер; графтар теориясы моделін қолданушы құрылымдық әдістер; машиналық оқыту теориясы негізінде құрастырылған әдістер және т.б. Әртүрлі тәсілдер жиынын меңгерген зерттеушілер олардың қайсысын таңдау керек деген қиындыққа тап болды, себебі олардың арасында айқын нақты «көшбасшы» болмады. Шындығында қарапайым, бірақ мәні барлық программалық қосымшаларға жарамды болатындай классификациялық модель жоқ екендігі туралы өте маңызды факт танылды. Бұл мәселе зерттеушілер тәжірибе жасау жолымен сәйкес тәсілді таңдап алуға тырысқан негізгі себептердің бірі болды. Ақыр соңында бұл әртүрлі классификаторлар жиынын біруақытта қолданудың артықшылығы бар екендігін түсінуге әкелді [11].

Бейне тану және классификациялау салаларында 1980-ші жылдарда классификациялау жүйесінің сапасын айтарлықтай арттыра алатын конструктивті жаңа идеялар жоқ, ал осындай идеяларды қажет ететін тәжірибелік қажеттіліктер үнемі өсу үстінде болды деп айтуға болады.

Классификациялау алгоримдерінің түрлері өте көп. Алгоритмдердің әрқайсысы бір деректер жиынын өңдеуде қолдануда жақсы нәтиже береді. Бірақ тура осы жиынға басқа бір алгоритмді қолданғанда, алдыңғы нәтижеден нашарырақ нәтиже алуымыз мүмкін. Осындай мәселелерді ескере отырып, бірнеше алгоритмдердің нәтижелерін топтастыру, яғни алгоритмдердің қорытында топтық шешімін алу осы диссертациялық жұмыста қарастырылады. Алгоритмдердің қорытынды топтық шешімдері негізінде, типтері әртүрлі деректерге кластерлік талдау жүргізе отырып, тиімді шешім аламыз. Алгоритмдердің топтық шешімін қолданудың артықшылығы:

1. Алгоритмдер жұмысының нәтижелерінің тұрақтылығы артады;
2. Топтаудың нақты бір қойылған шартын орындауда алынған нәтижелер сапасы артады;
3. Алгоритмдердің параметрлерін таңдаудан тәуелділік төмендейді;
4. Алынған шешімдердің сапасы жақсара түседі;

Топтық шешімдерді қабылдау мәселесі немесе топтық тану мәселесінің зерттелініп жатқандығына көп уақыт болды. Шамамен 1970 жылдан бастап осы уақытқа дейін топтық шешімдерді қабылдау мәселесі туралы жарияланымдардың саны тұрақты түрде артып, барлығының қызығушылық танытушы объектісі болып табылуда.

Деректерді талдау есебін топтық шешімдер алгоритмдерімен шешу идеясын практикалық түрде жүзеге асырудың теоретикалық негізі РҒА ЕО (ВЦ РАН)-да 1976-80 жж. тану есептерін шешу үшін және 1981-82 жж. кластерлік талдау есебі үшін классификациялаудың топтық синтез есебін шешу жасалынды. Уақыт өте келе осы бағытта көптеген елдерде жұмыстар атқарыла бастады. Қазіргі таңда бұл жасанды интеллект бағыты ретінде даму

үстінде. Бұл салада көлемі үлкен деректерге талдаулар жүргізіліп, көптеген қолданбалы есептерде пайдаланылуда.

Заманауи шет елдік әдебиеттерде топтық тану алгоритмдері мен әдістері әртүрлі ғылыми жұмыстарда әртүрлі атаулармен кездеседі:

- комитеттер (committees) [12,13];
- алгоритмдер ансамблы [14,15];
- эксперттерді біріктіру (mixture of experts) [16];
- классификаторлар жиынын біріктіру (combination of multiple classifiers);
- классификаторларды біріктіру (classifier fusion) [17];
- келісілген агрегация (consensus aggregation) [18];
- классификаторлар бірігуіне дауыс беру (voting pool of classifiers) [19];
- классификаторды динамикалық таңдау (dynamic classifier selection)[20];
- классификаторлардың аралас жүйесі (composite classifier system);
- аралас шешімдер (decision combining);
- «бөл де басқар» түріндегі классификатор (divide-and-conquer classifiers).

Шын мәнісінде, қолданылған терминдерде келтірілген әртүрлілік есеп қойылымдарының, тұжырымдардың және т.б. әртүрлілігін көрсетеді.

Топтық тану есебі бір немесе бірнеше алгоритмдер көмегімен жеке классификаторлардың шешімдерін келісе отырып, мағыналары бірдей кластар туралы шешімдерді әр қайсысы қабылдайтын классификаторлар жиынын қолдану есебі. Топтық тану есебі топтық шешімді қабылдаудың әртүрлі көптеген қосымшалардан тұратын жалпы мәселелеріне қатысты. Атап айтқанда, топтық шешімдер қабылдау есебіне топтық таңдау есебі, дауыс беру теориясы, эксперттік бағалауларды өңдеу [21] және т. б. жатқызуға болады.

Топтық шешімдер алгоритмдерін шешімдері ақырғы классификациялық шешімді алуда қандай да бір тәсілмен біріктірілетін (агрегацияланатын) жеке классификаторлар жиыны деп айтуға болады. Әдеби деректерге сәйкес классификаторлардың топтық шешімін пайдалану қолданбалы есептерді шешу кезінде классификатор дәлдігін арттыруға мүмкіндік береді [22]. Алайда, теориялық және эксперименттік зерттеулер топтық шешімдер алгоритмдерінің дәлдігінің қажетті және жеткілікті шарты оларды құрушы классификаторлардың әртүрлілігі мен олардың бір-біріне тәуелсіздігі екендігін көрсетеді [23]. Бұл шарт, біріншіден, бастапқы деректерді іріктеудің әр түрлі ішкі жинақтарында классификатоларды оқыту кезінде, екіншіден, классификаторлардың әртүрлі модельдерін топтық шешімдерде базалық ретінде(мысалы, ағаш түріндегі шешім, логистикалық регрессия, нерондық желі және т.б.) қолдану кезінде орындалады. Сондықтан, біртекті емес ансамбльдік алгоритмдерді әзірлеу және зерттеу бағытының болашы зор болып табылады. Осы бағытта ғылыми зерттеушілер саны уақыт өте ұлғая бастады. Бастапқы оқу жиынтығынан әртүрлі түрдегі базалық классификаторларды олардың әртүрлі ішкі жиындарында кездейсоқ үлгілерін қалыптастыру [24] жұмыста үшін бэгинг (bagging) топтық шешімдер

алгоритмдерін қолданатын біртекті емес топтық шешім алгоритмі көрсетілген. Мұнда аортты патологиясы бар науқастарды хирургиялық емдеу туралы нақты деректер пайдаланылған. Деректердің бастапқы таңдамасы екі бөлікке бөлініп қолданылған: оқытушы (классификаторларды оқыту үшін қолданылады) және тесттік.

Бейнені тану алгоритмдерінің топтық шешімі есебін бастапқы алгоритмдердің шешімдерінен тиімді нәтижелік бөлулер құру есебі. Тиімділік түсінігі классификациялаудың әр нақты есебі үшін анықталған. Дегенмен де бөлулердің нақты қиылысуы мүмкін болатын нұсқалардың көптеген бірнеше түрлерін қарастырып көруді қажет ететін өте күрделі мәселе болып табылады. Сондықтан бұл жерде базалық топтық шешім алгоритмдерінің жиынтығында тиімді нәтижелік бөлулерді құру туралы мәселе актуалды болып табылады. Танудың топтық әдісі әрқайсысы жеке нәтиже беретін бірнеше классификаторларды қолданады. Қандай да бір жалпы әр классификаторлардан алынған нәтижелер негізіндегі дауыс беру ережесі бойынша қорытынды нәтиже алынады.

Алгоритмдердің композициясын құру идеясы бұрыннан бері белгілі. Алгоритмдердің, алгоритмдер жиынының дауыс берулерінің әртүрлі мүмкіндіктері қарастырылған. Ең алғаш алгоритмдердің композициясының бірінші жалпы теориясын тануға алгебралық тәсілді қолданумен Ю.И.Журавлев және оның оқушылары ұсынды [25]. Дегенмен де бұл теория практикада аз қолданылды. 80-жылдардың соңында алгоритмдерді әлсіз және күшті оқытумен байланысты мәселелерді зерттеу жұмыстары пайда бола бастады. Зерттеу нәтижелері көрсеткендей күшті оқыту әлсіз оқытумен тепе-тең екендігін көрсетті, өйткені кез-келген әлсіз моделді дұрыс композиция құра отырып күшейтуге болады. 1996 жылы бұл идеялар қалыптастырылып, AdaBoost алгоритмі түріндегі аяқталған формасы алынды. Бұл алгоритм өзінің қарапайымдылығы мен тиімділігінің арқасында тез танымал бола бастады. Бірнеше жылдар өткен соң бұл алгоритмнің жалпыланған түрі – градиенттік бустинг алгоритмі пайда болды. Қазіргі кезде бұл ең танымал тану алгоритмдерінің бірі болып табылады. Бұл жетістікке алгоритмдер композициясын құрудың техникасының арқасында жетіп отыр. Бустинг алгоритмі вариация жасаудың көптеген мүмкіндіктерін береді, оны жоғалтудың әртүрлі функциясы ретінде қарастыруға болады. Мұны классификация есебі ретінде де, регрессия есебі ретінде де шешуге болады. Сонымен қатар жоғалтудың кез-келген функциясын таңдау мүмкіндігі есепте деректердің ерекшеліктеріне көңіл аударуға әкеледі.

Осы салада, яғни алгоритмдердің топтық шешімін қолдануда жасалған жұмыстар нәтижелік шешімдердің қандай тәсілдермен үйлестірілетіндіктерімен ерекшеленеді.

Топтық шешімдер алгоритмін құрудың негізгі әдістері:

- объектілердің жұптық ұқсастық/өзгешелік матрицасын қолдану;
- шешімдерді келісудің дәрежесін максимизациялау (Ранда түзетілген индексі және т.б.)
- теоретикалық графтық әдістерді қолдану;

Топтық шешімдер алгоритмін қолдану бағыты көптеген салаларда қолданылуда. Солардың ішінде бүгінгі таңда тұтынушылардың деректер қорынан банктік несиелерін өтей алмай қалу жағдайының қаупін болдырмауға мүмкіндік беретін скорингтік жүйелер мен әртүрлі алгоритмдер тобын құруға арналған жұмыстар көбейе түсуде. Осындай жұмыстардың біріне Ресей ғалымы И.А. Кузнецов пен В.С. Киреевтердің топтық шешім алгоритмдерін қолданумен құрылған кредиттік скоринг жасау туралы жұмысын жатқызуға болады [26]. Құрылған жүйе ерекшелігі нашар құрылымдалған деректер қолданылған. Жұмыста біртектілік өлшемінің орнына энтропиялық өлшем қолданылатын топтық шешім алгоритмдерінің жаңа нұсқасы жасалынған. Бұл жұмыста алгоритмдердің параметрлері мен салмақтары бірдей емес екендігін байқауға болады. Мұнда құрылымы нашар объектілермен жұмыс жасаудағы нәтижелеріне топтық шешімдер алгоритмінің актуалдық деңгейі жоғарырақтығын байқаймыз. Осыған ұқсас, алгоритмдердің топтық шешімдерін пайдалана отырып, деректерді интеллектуалды талдау аймағында ұсыну жүйесін жасау келесі авторлар Xiao Hongshan, Wang Yu-дың [27] жұмысынан көруге болады. Алгоритмдердің топтық шешімдерін қолданатын Bagging, Boosting алгоритмдері қолданылған және оларды құру жолдары көрсетілген. Кредиттік скоринг қаржылық нарыққа негізделген өңдеу платформаларында және қаржылық мекемелерде маңызды роль атқарады. Соңғы бірнеше жылдарда кредиттік мүмкінділігін бағалауда топтық шешімдер алгоритмдерін қолдану танымал бола бастады. Топтық шешімдер тәсілдерінің жақсы нәтиже беруі негізгі классификаторлар саны және ішкі параметрлеріне байланысты. Ағаш түріндегі шешімдер негізіндегі классификаторлардың топтық шешімін құруды компьютерлік желідегі қауіпті трафиктерді табу мәселесін қарастырған [28] – жұмысынан байқауға болады.

Классификациялау алгоритмдері, соның ішінде топтық шешімдердің нәтижелерін қолдану жерді қашықтықтан зондтауда (ЖҚЗ) кең қолданыста. Жерді қашықтықтан зондтау (ЖҚЗ) ғарыш қызметінің жылдам дамып келе жатқан саласы болып табылады. ЖҚЗ-ның ең перспективалы бағыттарының бірі инновациялық гиперспектральды әдістер мен технологияларды қолдану болып табылады. Қазіргі уақытта гиперспектрлік аэроғарыштық ақпарат табиғи ресурстарды зерттеу және ұтымды пайдалану, қоршаған ортаны қорғау, табиғи апаттар мен техногендік апаттардың алдын алу және зардаптарын жою, метеорология мен климатология, орман және ауыл шаруашылығы, көлік, жер туралы іргелі ғылымдар мүддесінде қалалық жоспарлау және басқа да көптеген міндеттерді шешу мүддесінде қолданылады. Бұл қызмет түрі инновацияларға аса бейім және іргелі және қолданбалы ғылымның ең соңғы жетістіктерін енгізуді талап етеді [29].

Бейне тану алгоритмдерінің топтық шешімдерін құру әртүрлі әдістерді бірге қолдануға мүмкіндік береді. Осындай шешімдерді алуда "әртүрлі көзқараспен" топтау жүргізіледі (бұл жерде "әртүрлі көзқарастар" тек бір-біріне қарама-қарсы емес, олар бір-бірінің кем тұстарын толықтырып отырады; шешімнің бір нұсқасы басқасының "әлсіз жақтарын" жауып отырады). Сонымен қатар кластерлер қалыптастырылумен сәйкес орнықты

заңдылықтар өзара “күшейе” түседі де, ал орнықсыз заңдылықтары керісінше “әлсірейді”.

Алгоритмдердің топтық шешімдерді бастапқы объектілер жиынын жеке кластарға бөлулерді келесі төменде көрсетілген жолдармен алуға болады:

1. Әртүрлі базалық алгоритмдер нәтижелері;
2. Бейне танудың таңдап алынған бір алгоритмінің баптауларын өзгертіп, алынған нәтижелерді топтап бір нәтижелік шешім алу арқылы.

Осы аталған жолдардың қайсысын қолдану талдау жасалушы деректер жиынының құрылымына байланысты.

Кластерлік талдаудың әртүрлі көптеген әдістерінің болуы топтық (немесе коллективтік, комиттеттік, ансамбльдік, келісілген) шешімдерін құру үшін осы әдістерді бірге қолдану мүмкіндігін береді. Осындай шешімдерді алуда “әртүрлі көзқараспен” топтау жүргізіледі (бұл жерде “әртүрлі көзқарастар” тек бір-біріне қарама-қарсы емес, олар бір-бірінің кем тұстарын толықтырып отырады; шешімнің бір нұсқасы басқасының “әлсіз жақтарын” жауып отырады). Сонымен қатар кластерлер қалыптастырылумен сәйкес орнықты заңдылықтар өзара “күшейе” түседі де, ал орнықсыз заңдылықтары керісінше “әлсірейді”. Соңғы уақыттарда көптеген авторлар [30] топтық шешімдерді қолдануға негізделген топтық шешімдердің тұрақтылығын арттыру тәсілдерін қарастыруда. Классикалық алгоритмдерде кластерлік талдау есептерін шешуде (мысалы, K-means алгоритмінде) топтау нәтижелері бастапқы шартты, объектілер ретін, алгоритмдер жұмысының параметрлерін және т.б. таңдауға байланысты айтарлықтай қатты өзгеруі мүмкін. Классификациялау алгоритмдерінің жиі қолданылатын алгоритмдер кластарының бірі бастапқы жиынның кейбір эталондық сипаттамаларын ерекшелеп алуға бағытталған алгоритмдер класы болып табылады. Әрекет ету қағидасына сәйкес мұндай алгоритмдер сонымен қатар кейбір эталондарды белгілеп алып, ары қарай осы эталондарға қатысты кластерлерді құратын эталондық алгоритмдер класына жатады. Мұнда эталондар ретінде қандайда бір жолмен белгіленіп алынған бірлік объектілер – болашақ кластардың центрі болып тағайындалатын кейбір бастапқы эталондарды сипаттаушы центрлер қолданылады.

Соңғы жылдарда кластерлік талдаудың топтық шешімдерін табуға арналған және оларды әртүрлі қолданбалы аймақтарда пайдалану жұмыстарының саны арту үстінде. Кластерлік топтық шешімдерді алудың келесі әдістемелері қарқынды түрде даму үстінде: шешімдердің сәйкестігін максимизациялау дәрежесі (Ранданың түзетілген индексі және т.б.); бутстреп-таңдама және т.б. Топтық шешімдер алгоритмдерін қалыптастыру әдістері көптеген жұмыстарда зерттелінген, мысалы [31, 32]-жұмыстарды.

Диссертациялық жұмыс мақсаты. Диссертациялық жұмыстың мақсаты бейне тану есептерінде құрылған тану және классификациялау алгоритмдер тобында жартылай бақылау арқылы оқыту және орталық объектілерді оқшаулау әдістері негізінде тиімді топтық шешімдер құрудың теориялық және тәжірибелік негіздеулерін ақпараттық жүйе құра отырып өңдеу және зерттеу.

Диссертациялық жұмыстың ғылыми жаңалығы: Жұмыстың ғылыми жаңалығы топтық шешімдер алгоритмдері және тану алгоритмдерімен де

жүргізілген диссертациялық зерттеу барысында алынған келесі ғылыми нәтижелерге негізделеді.

1. Классификациялаудың ядролық әдістерінің кластардың күрделі, сызықты емес шекараларын анықтау, сонымен қатар матрица ядросының рангісін азайту арқылы қажетті жады мен еңбек қарқындылығын төмендету мүмкіншіліктерін сәйкестендіре отырып, кластерлік талдау алгоритмдерінің көмегімен деректердің құрылымын нақты анықтау есебінен үлкен көлемді шулы, күрделі құрылымды деректерді талдаудың тиімділігін арттыруға мүмкіндік беретін топтық кластерлік талдау алгоритмдері және классификациялаудың ядролық әдістерін біріктіре отырып қолдану негізінде жартылай бақылау арқылы оқыту есебін шешуге арналған классификаторды құру алгоритмі зерттелінді және ұсынылды.

2. Таңдап алынған сапа функционалдары тобы бойынша тану есебінің дұрыс шешімін ұсынушы эталондық (ядролық) объектілерді оқшаулап алуға бағытталған классификациялау және тану алгоритмдерінің негізінде топтық шешімдердің тиімді алгоритмі зерттелінді және құрылды.

Зерттеу есептері:

1. Классификациялау және тану есептерінде топтық шешімдер алгоритмдерін зерттеу;

2. Классификациялау және тану есептерінде жаңа топтық шешімдер алгоритмдерін құру:

а) топтық тану мәселелерінің қойылуында жартылай бақылау арқылы оқыту есебін зерттей отырып шешу;

б) базалық алгоритмдер тобында орталық объектілерді оқшаулауға негізделген топтық шешім табу алгоритмі.

3. Топтық шешімдер алгоритмдерінің нәтижелерін талдау және бағалау.

4. Заманауи ақпараттық технологияларды қолдана отырып, жаңа топтық шешімдер әдістерінің нәтижелері негізінде танудың ақпараттық жүйесін құру;

Зерттеу объектілері. Объектілер жиыны, белгілер кеңістігі, жақындық метрикасы, кластар(кластерлер), тану және классификациялау алгоритмдері, сапа функционалдары, ақпараттық жүйелерді жобалау құралдары.

Зерттеу әдістері. Тану және классификациялау әдістері, жүйелік талдау және жүйелер теориясы, графтар теориясы, шешімдер қабылдау теориясы, программалық жабдықтарды өңдеу технологиялары.

Қорғауға ұсынылатын тұжырым. Топтық шешімдер табу есептерін шешуде жартылай бақылау арқылы оқытуға және орталық объектілерді оқшаулауға негізделген топтық шешімдер алгоритмдері бейне тану және классификациялау есебінің тиімді шешімдерін беретіні теориялық негізделіп, олардың тиімділігі мен шынайылығы ақпараттық жүйе аясында есептеу тәжірибелері арқылы көрсетілді.

Диссертациялық жұмыс нәтижелерінің апробациясы.

Диссертациялық жұмыстың орындалу барысында қол жеткізілген нәтижелер ғылыми нәтижелерге негізделген және олардың тиімділігі бейне тануда қолданылатын танымал алгоритмдерін қолдануда алынған нәтижелерімен салыстыра отырып расталады.

Диссертация нәтижелері әл-Фараби атындағы ҚазҰУ ақпараттық технологиялар факультетінің және ақпараттық жүйелер кафедрасының ғылыми семинарларында, сонымен қатар мына төмендегі ғылыми-әдістемелік конференцияларда баяндалып талқыланды:

1. Профессор Р.Г. Бияшевтың 80 жылдығына және профессор М.Б. Айдархановтың 70 жылдығына арналған «Информатика және қолданбалы математика» атты III Халықаралық ғылыми-практикалық конференциясында (Алматы, 26-29 қыркүйек 2018);

2. XIII Balkan Conference on Operational Research (BALCOR 2018) СЕРБИЯ Белград;

3. The 7 th International Conference on "Optimization Problems and Their Applications (OPTA-2018)" Russia, 2018;

4. Диссертациялық жұмысты орындаудағы алынған нәтижелер мен талдаулар бойынша 13 мақала жарық көрді және 1 авторлық куәлік алынды. Олардың ішінде Қазақстан Республикасы Білім және ғылым министрлігінің Білім және ғылым сапасын қамтамасыз ету комитеті ұсынған басылымдарда 4 (төрт), «Scopus» базасына енгізілген 4 (бес), халықаралық конференциялар материалдарында 5 (төрт) мақала жарық көрді.

Ғылыми жұмыстың жариялымдары.

1. Berikov V. B., Amirgaliyev Y.N., Cherikbayeva L.Sh, Yedilkhan D., Tulegeniva B. " Classification at incomplete training information: usage of group clustering to improve performance" Journal of Theoretical and Applied Information Technology. - 2019. - Vol.97. - № 19. – P. 5048-5060 *(Scopus базасы бойынша процентилі- 33).*

2. Amirgaliyev Y., Berikov V., Cherikbayeva L., Latuta K., Bekturgan K. "Group approach to solving the tasks of recognition" // Yugoslav Journal of Operations Research. - 2018. – Volume 2. – P. 177-192 (Scopus).

3. Sh. Shamiluulu, B. Y. Amirgaliyev, L. Cherikbayeva. " Critical analysis of scikit-learn ml framework and weka ml toolbox over diabetes patients medical data " // News of the National Academy of Sciences of the Republic of Kazakhstan, Series of Geology and Technical Sciences. - 2017. - Volume 6. - Number 426. - P. 231 – 236 (Scopus).

4. Berikov V., Cherikbayeva L. Searching for Optimal Classifier Using a Combination of Cluster Ensemble and Kernel Method // Optimization Problems and Their Applications (OPTA-2018), CEUR Workshop Proceedings, Omsk, Russia, Vol. 2098, – P. 45-60 (Scopus).

ҚР Білім және ғылым министрлігінің Білім және ғылым сапасын қамтамасыз ету комитеті ұсынған басылымда жарияланған мақалалар:

5. Амиргалиев Е.Н., Шамиль-улу Ш., Черикбаева Л.Ш., Кеншимов Ч.А. "О некоторых численных результатах распознавания с машинным обучением" // ҚазҰТЗУ хабаршысы. – 2017. -№ 2. – Б. 386-391.

6. Черикбаева Л.Ш. "Классификациялау және кластерлеу әдістері" // ҚазҰТЗУ хабаршысы. – 2017. -№ 2. – Б. 158-161.

7. Черикбаева Л.Ш., Байсылбаева Қ.Д. “Өзгермелі арақашықтық метрикасы негізіндегі алгоритмдер” // ҚазҰТЗУ хабаршысы. – 2018. -№ 2. – Б. 99-103.

8. Черикбаева Л.Ш. “Алгоритмдердің топтық шешімдерін пайдалана отырып тиімді классификаторларды іздеу” // Вестник КазНИТУ, - 2019. №2, – С. 289 - 292.

Халықаралық конференцияларында жарияланған мақалалар:

9. Kalimoldayev M., Amirgaliyev Y., Berikov V., Cherikbayeva L., Latuta K., Kalybek uulu B. One approach for the group synthesis of recognition and classification tasks // XIII Balkan Conference on Operational Research (BALCOR 2018), Belgrade. – P. 400-407 (Scopus).

10. Бериков В.Б., Амиргалиев Е.Н., Черикбаева Л.Ш. Полуконтролируемое обучение на основе кластерного ансамбля // Материалы IV Международной научно-практической конференции «Информатика и прикладная математика» 27-30 сентября 2017 года, Алматы, Казахстан, (Часть II), – С. 65-76.

11. Черикбаева Л.Ш., Калдыбекұлы Б. Кластерлік талдауда топтық шешудің тиімді параметрлерін таңдау алгоритмдері // Материалы IV Международной научно-практической конференции «Информатика и прикладная математика» 26-29 сентября 2018 года, Алматы, Казахстан, (Часть II), – С. 42-47.

12. Викентьев А.А., Серов М.С., Бериков В.Б., Черикбаева Л.Ш., Тулегенова Б.А. “Коллективные расстояние для кластеризации множеств формул N-значной логики”. // Материалы IV Международной научно-практической конференции «Информатика и прикладная математика» 25-29 сентября 2019 года, Алматы, Казахстан, (Часть I), – С. 219-234.

13. Черикбаева Л.Ш., Концепции построения распознающих и классифирующих систем // «Көліктегі инновациялық технологиялар: білі, ғылым, тәжірибе» атты XLI Халықаралық ғылыми-практикалық конференцияның материалдары, 3-4 сәуір 2017 ж., Алматы, Казахстан, (I том), 117-119 б.

«Software Semi-Supervised learning based on cluster ensenble» авторлық куәлігі 20.03.2019 ж. №6373.

Диссертация құрылымы мен көлемі. Диссертациялық жұмыс қазақ тілінде жазылды. Жұмыс кіріспеден, үш бөлімнен, қорытынды және қосымшалардан, әдебиеттер тізімінен тұрады. Диссертациялық жұмыстың жалпы көлемі 101-бет, 4-кесте, 47-сурет. Әдебиеттер тізімінде пайдаланылған дереккөздер саны 70.

Кіріспе бөлімде диссертациялық жұмыстың өзектілік мәселелері, мақсаты және оған жету үшін қойылған есептер қарастырылды. Қазіргі таңға дейінгі жасалынған тәжірибелер мен олардың нәтижелері, олардың ғылыми жаңалықтары сипатталған. Сонымен қатар тақырыпқа байланысты жарық көрген мақалалар тізімі берілді.

Бірінші бөлімде бейне тану әдістері, бейне тану алгоритмдері және олардың модификациялары қарастырылған. Алгоритмдердің негізгі

түсініктері, қағидалары көрсетілді. Алгоритмдер арқылы талдау жасап, зерттелуші объектілердің ұқсастық өлшемдерін анықтау қарастырылды. Сонымен қатар бұл бөлімде кластерлерді қалыптастыру тәсілдері және алгоритмдері, ядролық әдіске жататын тірек векторларының әдісі қарастырылды.

Екінші бөлімде топтық шешімдерге түсініктемелер, негізгі анықтамалар, белгілемелер беріліп, топтық шешімдер есебінің қойылымы келтірілді, сонымен қатар топтық шешімдер құрудың әдістері сипатталды. Жартылай бақылану есебіндегі топтық шешімдерге талдау жасалынды және қарастырылды.

Топтық шешімдер құрудың бірнеше концепциялары қарастырылды. Алынған топтық шешімдер алгоритмдері топтық шешімдер матрицасының енгізілген объектілі құрылымдық түсінігін қолданады. Орталық объектілерді – болашақ кластар эталондарын оқшаулауға және жұптық айырмашылықтар матрицасын негізделген топтық шешімдер әдістері, сонымен қатар класс аралық объектілі байланыстарды зерттеу келтірілді. Бөлімде бейнені тану есебінің қойылымы – жартылай бақылау арқылы оқыту есебі (semi-supervised classification) қарастырылып, топтық кластерлік талдау алгоритмдері мен классификациялаудың ядролық әдістерін үйлестіруге негізделе отырып берілген есепті шешуге арналған жаңа тәсіл алынды. Жартылай бақылау арқылы оқыту есебінде тек бастапқы таңдама объектілерінің бір бөлігі үшін ғана кластар белгілі болады; бұл есепте белгіленбеген объектілерді классификациялауға немесе жаңа объектілерді тануда шешуші ережелерді қалыптастыру кажет. Тәсілдің идеясы кластерлік алгоритмдердің топтық шешімдерін алынған коассоциациялық матрицаны объектілердің жұптық ұқсастық матрицасы ретінде алу болып табылады және оны нәтижелік бөлуде матрица ядросы ретінде қолданудан тұрады. Мұндай алмастырулар жасаудың бірнеше негіздемелері бар. Біріншіден берілген аймақтағы деректер құрылымы күрделі формада болғанымен де, объектілер бір-бірінен қандай да бір алыс ара қашықтықта орналасса да, әртүрлі кластерлерден алынған объектілерге қарағанда бір-біріне ұқсасырақ болып саналады. Екіншіден орталанған коассоциативті матрица бақылау кеңістігінде, объектілер жұбын бірдей кластерлерге жатқызу жиілігін сәйкес нүктелер арасындағы ұқсастық көрсеткіші ретінде қарастыруға болатындай жартылай метриканы анықтайды. Тарауда тексеру есептері мен нақты гиперспектральды кескіндермен жүргізілген сандық эксперименттер ұсынылып отырған әдістің, соның ішінде шулы деректердің болуы жағдайында тиімділігін көрсетеді. Көптеген жағдайда деректер құрылымындағы анықталмағандықтардың болуы орын алады. Мұндай жағдайда алгоритмдердің топтық шешімдерін пайдалану кластерлік талдау нәтижелерінің орнықтылығын арттыруға мүмкіндік береді. Бұл тарауда осындай тәсілдің қолданылуы орталанған коассоцативті матрицасының ұқсастық матрицасы ретінде қолданылуы шешім сапасының жоғары болатындығаң көрсетеді. Бұл бөлімде жартылай бақылау арқылы оқыту есебі қарастырылған. Бұл есепті шешуде гиперспектралды кескіндер пайдаланылып, оларға классификациялау жүргізілді. Классификациялау

нәтижелері қазіргі кездегі танымал топтық шешімдер алгоритмдерінің нәтижелерімен салыстырылды. Салыстыру нәтижелерінен ұсынылған алгоритмнің шуға орнықты екендігі, сонымен қатар дәлдігі, уақыт тиімділігі кесте түрінде көрсетілді. Мұнда оқыту процесі екі этаптан тұрады. Біріншіден кескінді сегменттерге бөлу нұсқасы құрылады. Ары қарай орталанған коассоциативті матрица есептелініп, келесі қадамда ұқсастықтары бойынша оқыту алгоритмдерін қолданылады. Осы арқылы кіріс дерегі ретінде есептелінген матрица қолданылатын шешуші функция алынады. Бөлімде алынған нәтижелердің танымал алгоритмдер нәтижелерімен салыстырылуы көрсетілді. Құрылған әдісті гиперспектралды кескінді талдау үшін қолдану мысалы сипатталған. Ұсынылған алгоритмнің шуға орнықты екендігі көрсетілген. Классификациялаудың топтық шешімдер алгоритмдерінің негізгі есебі қолданыстағы базалық алгоритмдер жиынынан алынған әрбір алгоритмдермен тиімді нәтижелік бөлуді құрудан тұрады. Мұнда тиімділік түсінігі нақты классификациялау есебі үшін таңдап алынған сапа функционалы бойынша келтіріледі. Сонымен қатар бөлімде ұсынылған әдіске жүргізілген эксперименттік зерттеулер және оның нәтижелері, алынған нәтижелердің танымал алгоритмдер нәтижелерімен салыстырылуы көрсетілді.

Үшінші бөлімде бейне тану және классификациялаудың ақпараттық жүйесін жобалау мен оны жүзеге асыру мәселесі қарастырылады. Ақпараттық жүйенің концептуалды схемасы келтірілді. Деректерді алдын-ала өңдеу мен енгізудің ішкі жүйесі, жүйені басқарудың ішкі жүйесі қарастырылды. Ақпараттық жүйені құруда алынған диаграммалар, топтық шешімдердің ішкі жүйесі көрсетілді. Құрылған жүйе зерттеушілерге объектілердің нақты бір анықталған тобын бейнені тану және классификациялау алгоритмдеріне сәйкес, соның ішінде топтық шешімдер алгоритмдерімен кластарға бөлуге мүмкіндік береді. Тану есебін шешу үшін топтық әдістер құрастырылды. Құрылған және жүзеге асырылған алгоритмдерден тұратын ақпараттық жүйе әртүрлі типті деректер (кескін, белгілер жиынымен сипатталған объектілер) үшін нақты тану және классификациялау есептері шешілетіндей платформаны ұсынады:

1. Бастапқы берілген деректер белгілі бір аумақтың спутниктік кескіндері түрінде ұсынылған (біздің мысалда бастапқы деректер ретінде «Ұлттық Ғылым Академиясы» мен «Сүлеймен Демирел атындағы университеттің» спутниктік кескіндері алынды). Берілген кескіннің объектілерін тану және классификациялау үшін кескін деректерін өңдеу қажет. Жақсартылған тануды алу үшін топтық шешім алгоритмі қолданылады. Жартылай бақылау арқылы оқыту негізінде кескіннің әртүрлі нұсқалары үшін (шуды ескере отырып) есептеу тәжірибелері жүргізілді.

2. Белгілер жиынтығымен сипатталған объектілер (үлгілер) бастапқы деректер ретінде ұсынылады. Бұл жерде объект ретінде Шу Іле аймағын гидрогеологиялық зерттеу кезінде алынған сынамалар алынды. Алынған объекттің физика-химиялық қасиеттері туралы зертханалық мәліметтері белгілер ретінде алынды.

Ақпараттық жүйе аясында ең жақсы нәтижеге қол жеткізу мақсатында

бастапқы деректерді өңдеу үшін классификациялау алгоритмдері (бустинг) жүзеге асырылды. Сонымен қатар бөлімде топтық шешімдер алгоритмдері арқылы алынған нәтижелер сапаларына бағалау жүргізу функционалдары қарастырылды.

Қорытындыда берілген диссертациялық жұмыстың негізгі нәтижелері мен қорытындылары сипатталды.

Алғыс білдіру. Автор ғылыми кеңесшісі т.ғ.д., профессор Әмірғалиев Еділхан Несіпханұлына ғылыми жұмысты орындауда көрсетілген көмегі, айтылған ескертулері, жан-жақты қолдау көрсеткендігі және де пайдалы кеңестеріне үлкен ризашылығын, шексіз алғысын білдіреді. Сонымен қатар шет елдік кеңесшісі Бериков Владимир Борисовичке де (Новосибирск қаласы, Ресей) көптеген пайдалы кеңестері мен диссертациялық жұмысты орындауда көрсеткен көмегі үшін алғысын білдіреді. Диссертациялық жұмыстың орындалуы кезінде айтылған кеңестері мен қолдаулары үшін әл-Фараби атындағы ҚазҰУ, «Ақпараттық технологиялар факультеті» және «Ақпараттық және есептеуіш технологиялар институтының» ұжымына, сонымен қатар диссертациялық жұмыстың орындалуда қолдау көрсеткені үшін отбасына шексіз алғысын білдіреді.

1 БЕЙНЕ ТАНУ ӘДІСТЕРІ МЕН АЛГОРИТМДЕРІ

Интеллектуалды ақпараттық жүйелер тұрғысынан қарағанда бейне – бұл деректерді талдау жасалынушы деректер жиынтығынан ерекшелеп алуға және қарастырылып отырған есептің шарттарымен сәйкес басқа объектілермен топтауға мүмкіндік беретін нақты немесе абстрактілі объектілердегі (процестердегі, құбылыстардағы) деректер жиынтығы.

1.1 Негізгі ұғымдар мен қағидалар

Нақты тану есебін шешуге қажетті объект сипаттамасы белгі деп аталады. Тану есептерінде нақты типтегі бірнеше объектілер тобын ерекшелеп алуға тура келеді. Бұл жағдайда әр топты басқа топтардан бөліп алуға мүмкіндік беретін белгілерді қолдану қажет. Мұндай топтарды тану есебінде кластар деп атау қабылданған. Жалпы жағдайда бейне ретінде нақты ақпараттық моделді алуға болады. Мұнда кездейсоқ алгоритмді бірнеше компоненттерден тұратын *E* абстрактілі функционалды жүйе ретінде қарастыруға болады:

$$E=\{D, F, Q\}$$

мұндағы, $D=\{D_s\}, s=1,...,L$ – категориялардың жиынтығы,

$F=\{S_k\}, k=1,...,m$ – сипаттамалардың жиынтығы,

$Q=\{Q_r\}, r=1,...,T$ – ережелердің жиынтығы (шешімдер қабылдаушы).

Жүйенің орындалуы үшін алдымен F жиынының элементтерінен тұратын баптау беріледі де, оған *Q* ереже орындалып, нәтижесінде *D* жиыны элементтеріне сәйкес белгі тағайындалады. *D, F* ақпараттық бөлігін білдіреді, ал *Q* – әдістемелік бөлігі болып табылады. Бұл жерден қорытындалай келе бейнені тану есебінің келесі тәсілдерін ерекшелеп атауға болады: эталондар арқылы салыстыру, кластеризациялау және сипаттамаларды жалпылау қағидасы. Әр қағидаларға сипаттама бере кетейік.

Эталондармен салыстыру қағидасы – есептеу құрылғыларының мүмкіншіліктері айтарлықтай шектеулі болған кездегі танудың техникалық жүйесін құруда пайда болған тәсілдердің бірі. Бұл тәсіл қазір де, атап айтқанда аналогтық және аналогтық-цифрлық тану жүйелерінде қолданылады.

Кластеризациялау қағидасы – егер де белгілер қандай да бір айқын емес өзара байланысты үлгіде берілген өлшемдер (параметрлер) жиынынан тұратын болса, онда бейнені белгілер кеңістігіндегі n-өлшемді вектор ретінде алуға болады. Әр класқа белгілер кеңістігіндегі векторлардың кейбір жиыны сәйкес келеді. Нәтижесінде класқа сәйкес келуші аймақтарға, яғни кластерлерге бөлінеді. Кластеризациялау қағидасы әртүрлі қолданбалы бағыттардағы сандық деректерді өңдеуде, атап айтқанда көпаймақты және спектраймақты аэрокосмостық бейнелерді (спектрлік белгілері бойынша классификациялау) компьютерлік талдау жүйелерінде кеңінен қолданылады.

Қасиеттерді жалпылау қағидасы бейне элементтері арасындағы байланыстарды пайдаланады. Ереже бойынша, ол эталондардың ақырғы санының сипаттамасын алу үшін әр кластың бейнелер жиыны өте үлкен

болған жағдайда қолданылады, бірақ бейнелердің ақырғы деректері бойынша кластардың өзгеше қасиеттерінің жеткілікті санын көрсетуге болады. Қасиеттерін көрсету сәйкес келетін модел негізінде жүзеге асырылады да, кейбір құрылым, функция немесе қатынас түрінде жадыда сақталады. Тану процесінде бейненің қажетті қасиеттерін көрсетуге мүмкіндік беретін сызба бойынша бейнені талдау жүргізіледі; содан кейін олар K_k кластары қасиеттерімен сәйкестендіріледі. Жалпылаушы қасиеттері ол бейнелерді түзуші алгоритмнің өзі де болуы мүмкін; бұл жағдайда бейнелер кластары нақты бір анықталған түрдегі құрылымды түзуші алгоритмдермен беріледі.

Соңғы жылдары деректерді талдау (Data Mining) бағыты даму үстінде. Бұл бағыт негізінде кластерлерді талдау алгоритмдерін қолдану арқылы жасалынған ғылыми жұмыстар көптеп орындалуда. Оны халықаралық баспа мақалаларынан, ғылыми конференциялар мақалаларынан және т.б. дереккөздерден көруге болады. Кластерлеу нәтижесінде объектілер жиынында ұқсас объектілер қандай да бір қойылған шарттар, параметрлер бойынша(бастапқы кластерлер саны) топтары бөлініп алынады. Деректерді талдау бағыты классификациялау есептерін шешуде ақпараттық жүйелерде, интернет-құжаттарды талдауда, кескіндерді сегментациялау және т.б. қолданылады. Деректерді талдауда болашағы зор кемел бағыттардың бірі кластерлік талдау болып табылады. Себебі осы күнде әлеуметтік, ғылыми және басқа да ақпараттық орталардағы жиналған көлемі үлкен деректерді талдауда тиімді кластерлік талдау әдістері қажет етіледі. Әртүрлі аймақтарда үлкен деректердің қарқынды түрде қолданылуы, зерттеушілердің үлкен көлемді ақпараттарды талдауы және өңдеу құрылғылары мен әдістерін дамытуға аса жоғары қызығушылығын арттырып отыр [33].

Кластеризациялау және тану есептерін шешу келесідей төмендегі қадамдардан тұрады:

1. Объектілерді белгілері бойынша ақпараттылығын анықтау. Ары қарай толықтығын анықтай отырып іріктеу жүргізу. Кластерге бөлінуші объектілерді таңдау.
2. Берілген объектілер жиынында классификациялау процестерін жүргізуге қажетті сипаттамалар таңдалынады. Сонымен қатар объектілердің ара қашықтықтықтарын анықтаушы тәсілдер анықталады.
3. Объектілердің бір-бірінен ерекшеліктерінің шектеуіш мәні есептелініп алынып, кластерлер қалыптастырылады.
4. Ақырғы кластерлерді алуда кластерлік талдау алгоритмін қолданылады. Талдау нәтижелерін көрсету немесе қандайда бір программалық кешен арқылы визуализациялау.

Мәліметтерді кластеризациялаудың негізгі этаптары 1.1-суретте графикалық түрде көрсетілген:

Бейне тану және классификациялау есептерінде объектілердің арақашықтықтарын анықтауда әртүрлі арақашықтық метрикалары қолданылады жән де осы арақашықтық метрикасының дұрыс таңдап алынса талдау нәтиженің сапасы артады.

Сурет 1.1 – Кластеризациялау этаптары

Кластеризациялау есептерінің шешімдердері бір мәнді болмайды, оның себептері бірнеше. Біріншіден, кластеризациялаудың сапасын анықтаудың әртүрлі критерилерінің болуы, сондықтан олардың барлығы әртүрлі нәтиже береді. Екіншіден, кластерлер саны алдын-ала белгісіз және кейбір субъективті критерилермен сәйкес анықталады. Үшіншіден, кластеризациялау нәтижесі таңдап алынған ара қашықтықты анықтау метрикасына байланысты. Төртіншіден, қолданылып отырған кластеризациялау алгоритміне байланысты. Жалпы кластерлеу келесі этаптарға бөлінеді: Кластерлеу объектілерін таңдау; Объектілер арасындағы ара қашықтықты өлшеу; Кластерлерді қалыптастыру; Нәтиже шығару. Ең бірінші есеп кластерлеуді қандай объектілері (айнымалылары) бойынша жүргізу қажетті екендігін анықтау керек. Осыдан кейін айнымалылар арасындағы жазықтықтағы ара қашықтық өлшемін есептеу керек. Айнымалылар арасындағы ұқсастықтар немесе айырмашылықтар таңдап алынған олардың арасындағы метрикалық ара қашықтықтарға тәуелді болады. Егер әр объект і қасиеттерімен (белгілерімен) сипатталса, онда ол i – өлшемді кеңістікте нүкте ретінде қарастырылуы мүмкін, сонымен қатар басқа объектілермен ұқсастықтары сәйкес ара қашықтықтар ретінде алынады. Ара қашықтық өлшемі қарастырылып отырған объектілердің ұқсастығын анықтайды. Ара қашықтық өлшемін есептейтін әдістердің бірнеше түрі бар, солардың ең танымал болып саналатындары Евклид арақашықтығы және Манхэттен арақашықтығы. Ара қашықтық есептелінген кейін кластерлеу алгоритмін қолданамыз. Деректердің құрылымы әртүрлі, сондықтан оларға сәйкес қолданылу ара қашықтық метрикасы болады. Шын мәнісінде, геометриялық тұрғысынан қарағанда, егер белгілер әртүрлі бірліктерде өлшенген болатын болса, онда евклид ара қашықтығын пайдалану дұрыс шешім бермеуі мүмкін [34,35]. Кластерлеу алгоритмдері, жалпы, екі категорияға бөлінеді – иерархиялық және иерархиялық емес. Иерархиялық алгоритм нәтижесінде дендрограмма құрылады, яғни, бұтақтары кластер болатын арнайы ағаш. Дендрограмма әдетте жақындық өлшемдерінің матрицасынан тұрғызылған ағаш ретінде түсіндіріледі. Дендрограмма берілген жиыннан объектілердің өзара қарым-қатынасын бейнелеуге мүмкіндік береді. Дендрограмманы құру үшін кластерлер арасындағы ұқсастық деңгейін анықтайтын ұқсастық (немесе айырмашылық) матрицасы қажет. Сіздің деректеріңіз үлкен болса және оның құрылымы бізге белгісіз болған жағдайда иерархиялық кластерді пайдаланған

ыңғайлы. Ұқсастықтарды анықтаушы ара қашықтық метрикаларының жиі қолданылатын түрлері 1.1-кестеде келтірілген.

Кесте 1.1 – Классификациядың арақашықтық метрикалары

Атауы	Формуласы	Ескерту
Евклид метрикасы	$P = \sqrt{\sum_{i=1}^{N}(A_i - B_i)^2}$	Евклид арақашықтығын келесі жадайларда қолданады: физикалық мағынасы бойынша объект қасиеті біртекті және классификациялау үшін маңызды болса; белгілік кеңістігі геометриялық кеңістігімен сәйкес келсе.
Евклид метрикасының квадраты	$P = \sum_{i=1}^{N}(A_i - B_i)^2$	Алыс ара қашықтықтағы объектілерге көбірек мән беруде қолданылады.
Салмақты евклид метрикасы	$P = \sqrt{\sum_{i=1}^{N} w_i\,(A_i - B_i)^2}$	Әрбір i-ші белгіге қандайда бір w_i салмағын беру жағдайында қолданылады. Салмақты анықтау, ереже бойынша, қосымша зерттеулермен байланысты, мысалы, эксперттерден ойларын сұрау және оларды өңдеуді ұйымдастыру.
Хэмминг метрикасы	$P = \sum_{i=1}^{N}(\lvert A_i\rvert - \lvert B_i\rvert)$	Манхэттен қалалық кварталдар ара қашықтығы деп те аталады. Бұл ара қашықтық координата бойынша айырмашылық болып табылады. Көптеген жағдайларда Евклид ара қашықтығындай нәтиже береді. Дегенмен де бұл өлшем үшін жеке үлкен айырмашылықтардың әсері азаяды.
Чебышев метрикасы	P =MAX$\lvert A_i\rvert - \lvert B_i\rvert$	Объектілердің сәйкес қасиеттерінің мәндері арасындағы айырмашылықтардың ең үлкен модулінің мәнін қабылдайды.

мұндағы P – А және В объектілерінің арасындағы ара қашықтық;

A_i – A объектісінің i – ші мәні;

B_i – B объектісінің i – ші мәні;

Кластеризациялаудың базалық алгоритмдері берілген объектілер жиынын қиылыспайтын ішкі жиындарға бөледі. Сондықтан кез-келген объект тек бір ғана кластерге тиісті болады.

Кластерлік талдаудың кең таралған әдістері мен тәсілдерін шартты түрде екі топқа бөлуге болады:

1. Жиын нүктелері арасындағы ара қашықтықтағы берілген шектеуде кластерлерді анықтау (арттыру) әдісі

2. Топтардың берілген саны бойынша кластерлерді қалыптастыру.

Бірінші тәсілде кластерлер саны белгісіз. Мұндай есептер қойылымында бастапқы мәліметтер ара қашықтық шекаралары болып табылады, ал алынған нәтиже бастапқы шарттарға тәуелділігін байқаймыз.

Екінші тәсілде бастапқы параметрі берілуі керек, олар кластерлер саны да болуы мүмкін. К-орталар тобы алгоритмін екінші тәсілге жатқызуға болады. Осы тәсіл алгоритмдеріне көп қызығушылық танытуда, сонымен қатар қазіргі уақтта көп қолданысқа ие. Сондықтан да төменде осы алгоритмдер тобын қарастырдық.

Осы және басқа да тәсілдерде кластерлерге қосымша талаптар қойылуы мүмкін: топта нүктелердің санының аз болуы, топтар арасындағы минималды ара қашықтық және басқа да. Төменде қарастырылған алгоритмдердің көпшілігі объектілердің ұқсастығын бағалауда евклид метрикасына негізделген. Мұндай метриканы қолдану жақындықтың жалпы қабылданған концепциясымен жақсы сәйкес келеді.

Көптеген графтар теориясына негізделген кластеризациялаудың тиімді әдістері бар. Олардың кейбіреулерімен [36] жұмысынан танысуға болады.

1.2 Кластерлерді анықтау (өсіру) әдісі. Толық байланыс алгоритмі

Бастапқы бейне ретінде қандайда бір «шеткі» мысалы, X кеңістігіндегі минималды кординаталы нүктені алайық,. Оны K_1 кластерінің m_1 центрі деп атайық. K_2 екінші центр ретінде барлық бейнелер жиыны бойынша айтарлықтай алыстатылған нүктені таңдайық. d шектік мәнді келесі формуламен анықтайық:

$$\mathrm{d} = \|\mathrm{m}_1 - \mathrm{m}_2\|/2$$

1-қадам. Алдымен біздің бейнелер жиынынан барлық x үшін m_1 және m_2 центрлеріне дейінгі ара қашықтықтарды $||x-m_1||$, $||x-m_2||$ есептейміз. Әр ара қашықтық жұбынан минималды ара қашықтықты таңдаймыз.

2-қадам. Барлық бейнелер жиынынан

$$M=\max\{\min(||x-m_1||,||x-m_2||)\} \qquad (1.1)$$

максималды мәнін есептеп, анықтап аламыз. Осы мәнге x_i бейнесі сәйкес келсін. Егер M >d болса, x_i мәнін $К_3$ кластерінің центрі деп тағайындаймыз. Жаңа шектік өлшемі d ретінде d=M/2 шамасын алуға болады.

3-қадам. Барлық x үшін түзілген кластардың центрге дейінгі K ара қашықтықтарының минималды мәнін есептейміз: $min||x-m_k||$, $k=1,...,K$.

4-қадам. Центріне дейінгі x бейнесінің $\rho_{opm}(x,m_k)=||\ x-m_k||/N$ орташа минималды қашықтығын есептейміз. Мұндағы, N - (x,m_k) жұбының жалпы саны. Жаңа $d=\rho_{opm}(x,m_k)$ шегін тағайындаймыз.

5-қадам. Барлық бейнелер жиыны бойынша (1.1)-теңдіктен М мәніне сәйкес келетін x_i мәнін іздейміз. Егер $M<=d$ болса процесс аяқталады. Кері жағдайда x_i мәнін кластердің кезекті центрі ретінде тағайындаймыз да 3-қадамға өтеміз.

Процесті d шамасы біз жұмыс жасап отырған белгілер өлшемінің орташа квадраттық қателігінен кіші болған жағдайда да тоқтатуға болады. Берілген алгоритм бейнелердің шағын тобын алуға мүмкіндік беруіне қарамастан басқа кластерлерді анықтау алгоритмдері сияқты ол да эвристикалық болып табылады. Объектілі нәтижеге қол жеткізу үшін кластеризациялау сапасының математикалық негізделген көрсеткіштерін қолдану қажет. Бұл сапа критерилері кластеризациялар орындалатын нақты есептер жоспарымен келісуге болады.

1.3 Топтардың(кластар) саны алдын-ала берілген жағдайда кластерлерді қалыптастыру алгоритмдері

Кластерлеу алгоритмдерінің саны өте көп екендігі белгілі. Кластерлеу кезінде кластерлердің топтардың(кластар) саны, кластер орталары, арақашықтық метрикасы сияқты параметрлер қолданылады. Кластерлеуде топтардың(кластар) саны нақты алдын-ала берілген жағдайда кластерлерді қалыптастыру алгоритмдері тобына жататын танымал алгоритм – К-ішкі топтық орталар алгоритмін айтуға болады.

1.3.1 К - ішкі топтық орталар алгоритмінің модификациясы

Бейне тану алгоритмдерінің кең таралған түрлеріне сапа көрсеткіші ретінде орташа квадраттық қателік минимумын қолданушы алгоритмдер жатады. Көптеген жағдайларда евклидтік өлшем орнына нәтижеге ешқандай әсер етпейтіндей, есептеуді айтарлықтай жеңілдететін жай ғана арақашықтық квадраты қолданылады.

Осындай алгоритмдердің негізі К-ішкі топтық орталар алгоритмі бойынша кластеризациялау болып табылады. бұл алгоритм көмегімен берілген объектілер жиынының элементтерін кластерлеуден кейін пайда болған кластерлер орталарын қадам сайын итеративті түрде дұрыстап отыру, яғни ішкі топ орталарының квадраттарының қосындысын минимизациялау процесі жүргізіледі.

1-қадам. Барлық берілген объектілер К кластерінде келесі шарттар бойынша үлестіріледі:

$$||x-m_k||\rightarrow\min$$

яғни объектілер арақашықтықтары min-ға ұмтылуы керек, мұндағы m_k – кластерлер ортасы.

2-қадам. 1-қадам нәтижесі бойынша m_k кластерлерінің жаңа орталары есептелінеді. Сонымен қатар $||x-m_k||$ арақашықтықтар квадраттарының

қосындысы *к*-шы кластерден барлық х бойынша минимум болуы керек деп ұйғарылсын. Бұл жағдайда m_k центрі берілген кластер үшін таңдаулы ортасы болып табылады:

$$m_k = \frac{1}{N_k}\sum_{k=1}^{N_k} x_j \qquad (1.2)$$

мұндағы N_k – *k*-кластердегі *x* бейнелер саны. Осыдан да берілген алгоритм атауы алынған.

3-қадам. Егер де 2-қадамда есептелінген жаңа орталар алдыңғы итерацияда есептелінген центрлермен сәйкес келетін болса (немесе берілген ε шамасынан артық ерекшелігі болмаса) процесс аяқталады. Әйтпесе 1-қадамға ораламыз.

К-ішкі топтық орталар алгоритмінің нәтижесі алдын – ала таңдап алынатын параметрлерге, мәліметтердің құрылымдық ерекшеліктеріне тәуелді болады. Тиімді нәтижелерді мәліметтер құрылымы бір-бірінен айтарлықтай алшақта қалып қоятын сипаттамалық топты құрған жағдайда алуға болады. Егер де мұнда анық көрсетілген нүктелердің шоғырлану аймағы болмаса кластеризациялау процесі ұзаққа созылып кетуі мүмкін және бұл жағдайда деректердің үлкен массивін өңдеу кезінде ескерген жөн болады.

1.3.2 Деректерді талдаудың интерактивті өздігінен құрылатын әдісінің бір нұсқасы

Берілген алгоритм кластеризациялау процесін айтарлықтай жетілдіре түсетін қосымша көптеген параметрлерден тұрады. Кластеризациялау процедурасы келесі үш негізгі блоктардан тұрады:

1) Центрлер аппроксимацияларының блогы;
2) Центрлерді бөліктеу блогы;
3) Центрлерді біріктіру блогы;

Жеке блокта осы көрсетілген процестердің біреуі аяқталған кезде басқаруды тасымалдайтын процедураны жасай аласыз.

Мұндай алгоритмнің жұмыс жасауы үшін келесі төменде көрсетілген параметрлерді беру қажет.

K – кластерлердің бастапқы саны,

N – қажетті кластерлердің саны,

$(m_1,...,m_K)$ – кластерлер орталары,

θ_s – объектілер санына төменнен шектеу,

θ_σ – ауытқуға төменнен шектеу,

θ_c – орталардың ара қашықтықтарына жоғарыдан шектеу,

L – біріктіру блогындағы орталардың саны,

I_{max} – итерация.

1. *Аппроксимациялау(орталарды).*

1-қадам. Объектілерді кластерлер бойынша үлестіру:

$$||S\text{-}m_p||=min||\ S\ \text{-}m_k||,\ k=1,...,K\ \rightarrow\ x\epsilon\Omega p$$

2-қадам. Ұсақ кластерлерді келесі шарт бойынша тексеріп, алып тастау:

$M_k<\theta_s \rightarrow m_k$ - ны алып тастаймыз, орталар санын кішірейтеміз: $K=K$ – кластерлеуде өзгерістер орын алса бірінші қадамға қайтамыз. Егер кластерлер азаятын болса бірінші қадамға қайтамыз.

3-қадам. Алынған нәтижелерден жаңа орталарды есептеу керек. Ол үшін (1.2)-ші формуланы пайдаланамыз.

4-қадам. Орташа арақашықтықты есептеу

$$G_{opm}(k)=\frac{1}{M_k}\sum ||x_i - m_k||,\ i=1,..., N_k,\ k=1,...,K$$

5-қадам. Мұнда жалпыланған орташа арақашықтық табылады:

$$G = \frac{1}{K}\sum_{k=1}^{K} M_k G_{\text{орт}}\,(k)$$

6-қадам.

1) Егер итерация берілген бастапқы итерацияға теңессе онда $I=I_{max}$, онда $\theta_c=0$. Бұл жағдайда 12-қадамға өтеміз (топтарды біріктірмей процесті аяқтаймыз); $\theta_c=0$ нөлге теңестіру біріктіру блогынан автоматты түрде шығып кетумен қамтамасыз етеді, өйткені бұл жағдайда біз біріктіру үшін кластерлер жұбының бірін де таба алмаймыз;

2) $K\leq {}^{N}/_{2}$ → 7-қадамға қайтамыз;

3) Егер I – жұп және $K\geq 2N$ болса, онда алдымыздағы 12-қадамға өтеміз (кластерлерді біріктіру);

4) 1-3 шарттары орындалмаса, онда m_k орталарының жаңа мәндерін қолданамыз. Ары қарай 1-қадамға қайтап, центрлерді іздеуді жалғасытрамыз

2. *Кластерлердің орталарын ерекшелеп алу*

Бұл бөлімде кластерлердің қаншалықты созылғанын және қай бағытқа созылуы шамамен есептеледі.

7-қадам. Әр K кластерлер үшін $\sigma_к$ орташа квадраттық ауытқу векторы есептелінеді. Бұл вектор компоненттері S кеңістігінің n-өлшемді координаталары бойынша келесі формуламен есептелінеді:

$$\sigma_{jk} = \sqrt{\frac{1}{N}\sum_{x\epsilon\Omega_k}(S_{ik} - m_{ik})^2}\ ,\ k=1,..., K,\ j=1,...,n$$

8-қадам. Әр $k=1,..., K$ кластер үшін S кеңістігінің барлық координаталары бойынша максималды компоненттері анықталады (яғни максималды таралу бағыты)

$$\sigma_{max}(k)=max\sigma_{jk},\ j=1,...,n$$

9-қадам. Егер барлық таңдалынған $\sigma_{max}(k),\ k=1,\ldots,\ K$ үшін $\sigma_{max}(k)>\theta_\sigma$ шарты орындалса, онда келесі қадамға өтеміз, әйтпесе 12-қадамға өтеміз.

Мұнда 10 және 11-қадамдар циклде барлық $k=1,\ldots,K$ кластерлер бойынша орындалады.

10-қадам. Келесі шарттарды тексеру

а) $D_{opm}(k)>D$ *және* $N_k>2(\theta_s+1)$

б) $K\leq N/2$

Шарттардың бірі орындалса, 11-қадамға ораламыз, әйтпесе $k=k+1$ болады (келесі кластерді тексеруге өтеміз).

11-қадам. 10-қадам шарты орындалатын k-шы кластерді бөлеміз.

Мұнда бөлу қандай да бір $0<\gamma\leq 1$ шамасы таңдап алынады және mk центрінен j координатасы бойынша ерекшеленетін екі жаңа центрлер түзе отырып, ең үлкен компоненті бойынша жүргізіледі.

$$m_{jk}^{+}=m_{jk}+\gamma\sigma_{max}(k),\ m_{jk}^{-}=m_{jk}-\gamma\sigma_{max}(k)$$

мұндағы γ шамасы айтарлықтай басқа кластерлердің күйіне әсер етпейтіндей таңдап алынады.

3. *Центрлерді біріктіру*

12-қадам. Барлық кластерлер жұптары арасындағы D_{ks} арақашықтықтары есептелінеді:

$$D_{kl}=||m_k-m_l||\ ,\ k=1,\ldots,K,\ l=1,\ldots,K,\ k\neq s$$

13-қадам. Барлық жұптар үшін $D_{kl}<\theta_c$ шартының орындалуы тексеріледі. Бұл шарт орындалатын жұптар D_{ks} өсуі бойынша реттеледі және алғашқы L жұп – кандидаттар біріктіру үшін таңдап алынады. Егер мұндай жұптар жоқ болса, 15-қадамға өтеміз.

14-қадам. Біріктіру жұп бойынша жүргізіледі. Әр *L* жұбы үшін жаңа центрдің есептелінуі:

$$m_p^{*}=\frac{1}{N_k+N_l}(N_k m_k+N_l m_l)$$

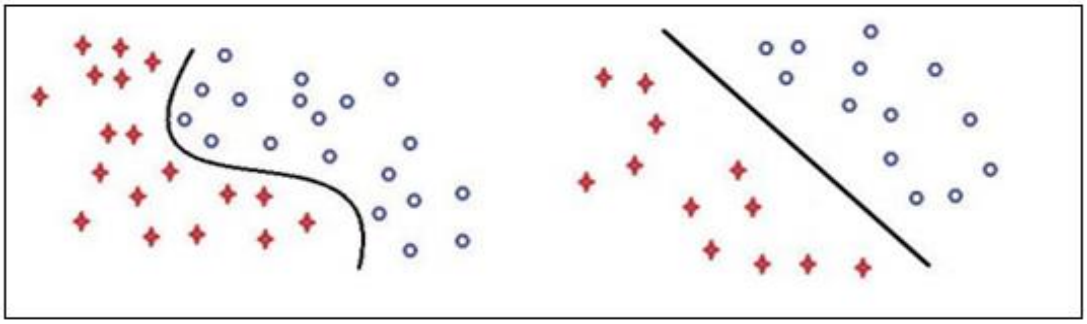
мұнда m_k және m_l центрлері біріктіріліп, сәйкесінше кластерлердің саны да кемиді.

15-қадам. $I=I_{max}$ болса, процесс тоқтатылады, әйтпесе 1-қадамға ораламыз. Үлкен көлемді деректердің өңделуінде бұл алгоритмнің басқа да түрлері қолданылады.

1.4 Тірек векторларының алгоритмі

Әрбір классификатор (классификация алгоритмі) келесі бейнені анықтайды: $A:Z\rightarrow Y$. Кластар арасындағы шекараны бөлуші бет немесе гипержазықтық деп атайды. Объектілерді дұрыс бөлуші гипержазықтық тұрғызу үшін әртүрлі тәсілдерді пайдалануға болады. Класс пен гипержазықтық арасындағы алшақтық – ол класс объектілері мен

гипержазықтық арасындағы минималды алшақтық. Онда тиімді гипержазықтық – осы алшақтықты максимизациялаушы гипержазықтық болып табылады. «Алшақтығы кең» классификатор ретінде тірек векторларының алгоритмдерінің бірі SVM (Support Vector Machine) болып табылады.



Сурет 1.2 – Кеңістіктегі бөлуші қисықтардың мысалы

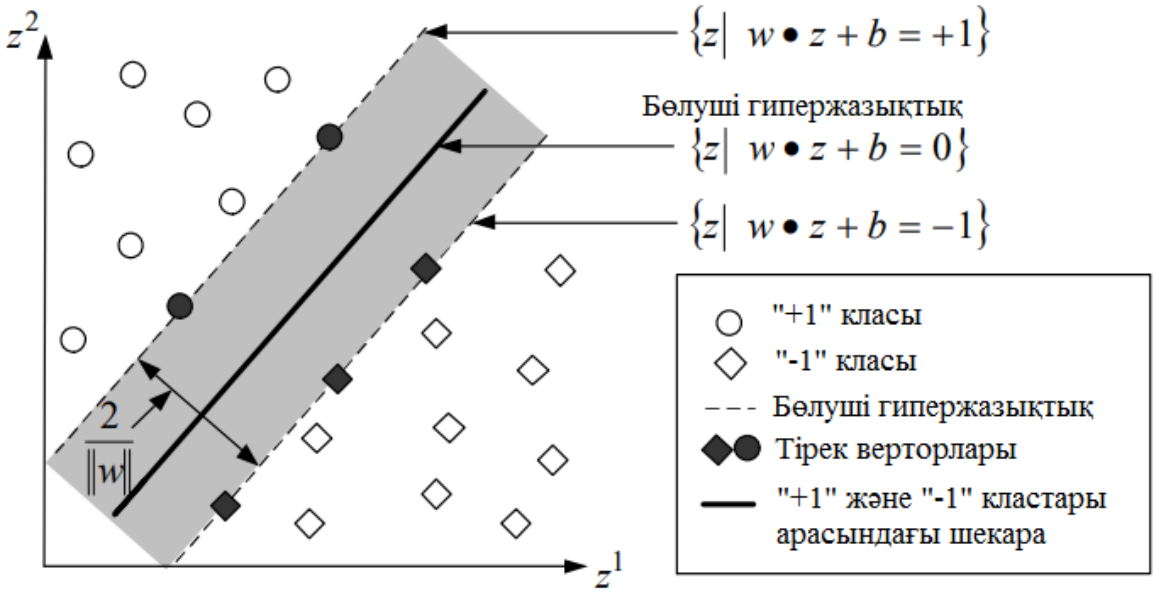
Тірек векторлары әдісінің мақсаты тиімді бөлуші гипержазықтықты құру болып табылады. Бөлуші жазықтықтардың шетінде жатқан нүктелер тірек векторлары деп аталады. Келесі $\langle w, x\rangle + b = 0, \langle\,,\rangle$ - скалярлық көбейтінді, w-бөлуші гипержазықтыққа перпендикуляр вектор, ал b-қосымша параметр. Параллель бөлуші гипержазықтықтың шекаралық түзулерін келесі теңдеулермен жазуға болады:

$$\langle w, x\rangle + b = \quad 1; \qquad \langle w, x\rangle + b = -1$$

Берілгендердің сызықтық бөлінуі туралы тұжырым бойынша бұл түзулерді олардың арасында бірде-бір нүкте жатпайтындай етіп таңдап аламыз. Енді біз бөлуші жолақтың енін максимизациялау керекпіз (шекаралық түзулердің ара қашықтығы). Ол $\frac{2}{\|\omega\|}$ -ға тең. Шын мәнісінде гипержазықтық теңдеуі келесідей болады:

$$\frac{w}{\|\omega\|} + \frac{b-1}{\|\omega\|}=0;\ \frac{w}{\|\omega\|} + \frac{b+1}{\|\omega\|}=0;$$

мұнда $\frac{b-1}{\|\omega\|}$ және $\frac{b+1}{\|\omega\|}$ – осы гипержазықтықтардан координата басына дейінгі арақашықтық, $\frac{2}{\|\omega\|}$ – гипержазықтықтар арасындағы арақашықтық. Яғни, бұл жерде $\frac{2}{\|\omega\|}$-ні максимизациялау керек немесе $\|\omega\|$-ді минимизациялау керек. Бөлуші гипержазықтық ені үлкен болған сайын нәтиже жақсырақ болады.



Сурет 1.3 – Кластарды гипержазықтықтармен бөлу

Келесі оптимизациялау есебін аламыз:

$$\begin{cases} \arg\min\limits_{w,b} \|w\|^2, \\ y_i(\langle w, x_i\rangle + b) \geq 1, i = 1, \dots, n. \end{cases} \tag{1.3}$$

Кун-Такер теоремасы бойынша (1.3)-оптимизациялық есебін шешу лагранжианның ершік нүктесін іздеумен эквивалентті болып келеді:

$$\begin{cases} L(w, b;\ \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i (c_i((\langle w, x_i\rangle + b) - 1)) \to \min\limits_{w,b} \max\limits_{\lambda}, \\ \lambda_i \geq 0, 1 \leq i \leq n. \end{cases}$$

Бұл теңдеуді шешуден w, b- келесі формулаларымен анықтаймыз:

$$w = \sum_{i=1}^{n} \lambda_i c_i x_i\ ;$$

$$b = \langle w, x_i\rangle - c_i, \lambda_i > 0$$

$$\phi(x) = sign(\sum_{i=1}^{n} \lambda_i c_i \langle x_i, x\rangle + b)$$

Бұл жерде қосындылау тек тірек векторы бойынша ғана жүретінін айта кету керек, $\lambda_i \neq 0$. $\phi(x) = 1$-ден $x \in X$ объектілері бір класқа, ал $\phi(x) = 0$-ден басқа кластарға жатады. $\phi(x)$ шешуші функциясы объектілердің өзінен емес $\langle , \rangle$ скаляр көбейтіндісінен тәуелді. $\langle x, x' \rangle$ скалярлық көбейтіндісін кеңістікте $\langle \phi(x), \phi(x') \rangle$ көбейтіндісіне ауыстыруға болады. Онда $\phi(x)$ келесі түрде болады:

$$\phi(x) = sign(\sum_{i=1}^{n} \lambda_i c_i \langle \Psi(x), \Psi(x') \rangle + b)$$

K(x, x')= $\langle \Psi(x), \Psi(x') \rangle$ ядро деп аталады. Скалярлы көбейтіндіден кездейсоқ ядроға өту ядролармен амал жасау («kernel trick») деп аталады.

Бөлім бойынша тұжырым

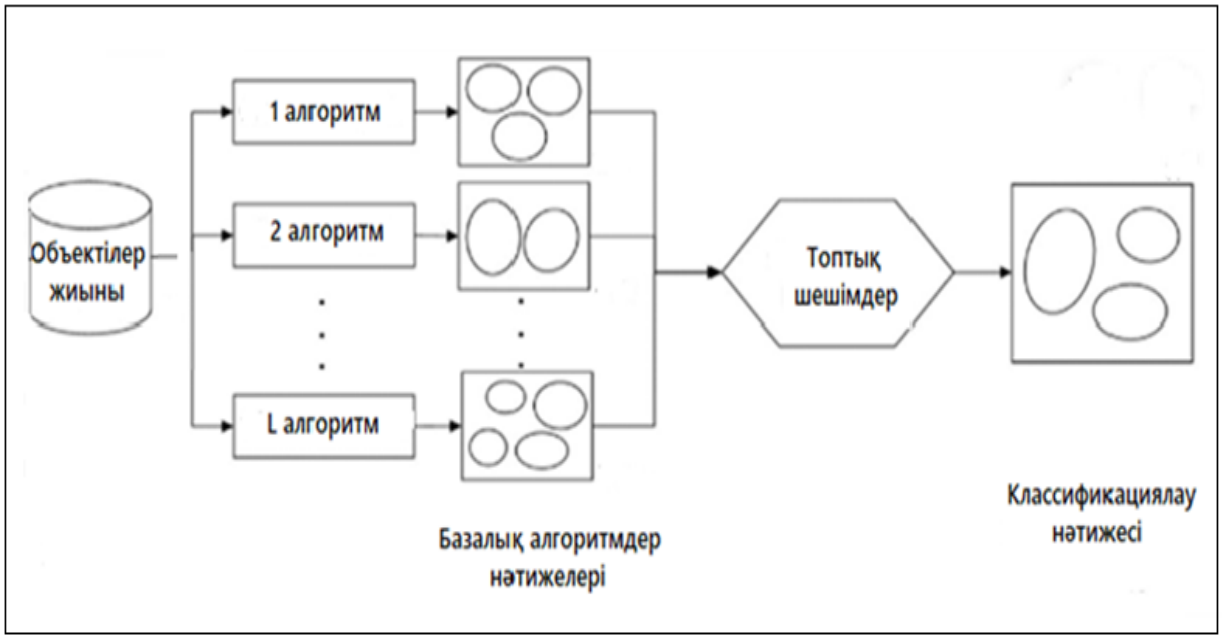
Берілген бөлімде бейнені танудың кең таралған алгоритмдері және олардың модификациялары және бейне тану процесіндегі шешім қабылдаудағы негізгі бағыттар қарастырылды. Сонымен қатар бейне тану есептерін шешудегі негізгі бағыттары мен әдістері сипатталып, берілген объектілердің ұқсастықтарын анықтауда қолданылатын ара қашықтық функциялары қарастырылды. Бұл жерде машиналарды оқытудағы әдістердің бірі - ядролық әдістерге жататын тірек вектор әдісі (SVM) қарастырылды. Бұл – бейнені тану алгоритмдерінің бір класына жатады және олардың ең танымал өкілі болып табылады. Ядролық әдістер деп аталуы ядролық функцияларды қолдану арқасында пайда болды, олар кеңістіктегі деректердің координаттарын есептеместен, кең белгілер кеңістігінде жұмыс істеуге мүмкіндік береді, белгілер кеңістігіндегі барлық деректер жұптарының кескіндерінің арасындағы скаляр көбейтінділерді есептейді. Бұл операция анық координаталық есептеулерге қарағанда жиі арзан болады. Мұндай тәсіл «ядролық трюк» деп аталады. Әдістің негізгі идеясы - бастапқы векторларды жоғары өлшемді кеңістікке ауыстыру және осы кеңістіктегі максималды алшақтықпен бөлетін гипержазықтықты іздеу.

2 БЕЙНЕ ТАНУ ЕСЕПТЕРІНІҢ ШЕШІМІН ТАБУДА ТОПТЫҚ ШЕШІМДЕРДІ ҚОЛДАНУДЫ ЗЕРТТЕУ ЖӘНЕ ҚҰРУ

Қазіргі таңда бейне тану есептерін топтық шешімдер арқылы шешу бейне танудың жеке алгоритмдері арқылы шешуден алынған нәтижелерге қарағанда жақсырақ нәтижелер беретінін шет елдік, отандық ғалымдардың ғылыми жұмыстарынан, әдебиеттерден көреміз. Топтық шешімдер арқылы кластерлік талдау жасау жеке алгоритмдер нәтижелеріне қарағанда анағұрлым жақсы нәтижелер беруде және бір уақытта бірнеше алгоритмдердің артықшылықтары мен ерекшеліктерін қолдануға мүмкіндік береді.

2.1 Топтық шешімдер алгоритмдерінің анықтамалары мен белгілеулері

Бейнелерді тану есебі қандай да сапа критерилеріне сәйкес бастапқы берілген объектілер жиынын бірнеше кластарға бөлу болып табылады. Нәтижелік бөлуді алу үшін қолданылатын алгоритмдер өте көп, сонымен қатар топтық шешімдер алгоритмдері арқылы да алынады. Нәтижелік бөлуді алуда бейне тану және классификациялаудың базалық алгоритмдер жиыны алынады да, олардың әрқайсыларының шешімінен қойылған талаптар, шарттарды қанағаттандыратындай шешімдері іріктеп алынып, осы шешімдер негізінде топтық шешім алынады (2.1–сурет).



Сурет 2.1 – Топтық шешімнің құрылу сызбасы

Топтық шешімдер алгоритмін қарастырайық. Айталық $S=\{S_1,S_2,S_3,\ldots,S_m\}$ объектілер жиыны берілсін және де жиындағы әрбір объект әрбір объект төмендегідей белгілерден тұратын болсын:

$$J_m(S) = \left\| \begin{matrix} a_{11} & \dots & a_{1d} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{md} \end{matrix} \right\|_{mxd}$$

мұндағы, d-белгілер, m-объектілер саны, $S_i = \{ a_1, a_2, a_3, \dots, a_m \}$, $S_i \in R^d$.

Берілген объектілер тобын l кластерлер тобына бөлу есебін қарастырамыз.

Анықтама 1. Берілген $S = \{S_1, S_2, S_3, \dots, S_m\}$ объектілер жиынын $\{K_1, K_2, \dots, K_l\}$ кластарға бөлуді 1-типтегі кластеризациялау деп атаймыз, егер алынған объектілер жиынының жүйесі S жиынының бөлінуі болса, яғни $\bigcup_{i=1}^{l} K_i = S$, $K_i \cap K_j = \emptyset$, $i \neq j$. 1-типтегі кластеризациялаудың барлық жиынын $\{K\}_1$ арқылы белгілейік.

Анықтама 2. Берілген $S = \{S_1, S_2, S_3, \dots, S_m\}$ объектілер жиынын $\{K_1, K_2, \dots, K_l\}$ кластарға бөлуді *2-типтегі кластеризациялау* деп атаймыз, егер алынған объектілер жиынының жүйесі жиынының жабынуы болса, яғни $\bigcup_{i=1}^{l} K_i = S$.

2-типтегі кластеризациялаудың барлық жиынын $\{K\}_2$ арқылы белгілейік. 1-типтегі кластеризациялау 2-типтегі кластеризациялаудың дербес жағдайы болып табылады.

Анықтама 3. $I = \|\alpha_{ij}\|_{m \times l}$ матрицасы $S = \{S_1, S_2, S_3, \dots, S_m\}$ объектілер жиынын $\{K_1, K_2, \dots, K_l\}$ кластарға 2-типті кластеризациялауда алынатын *ақпараттық матрицасы* болып табылады. Бұл жерде матрицаның өлшемі $m \times l$-ға тең.

Анықтама 4. Ақпараттық матрица *эквивалентті* болып табылады, егер $I = \|\alpha_{ij}\|_{m \times l}$ және $I' = \|\alpha_{ij}'\|_{m \times l}$ тең болса.

Тұжырым: Егер де $\|T_{ij}\|_{m \times m}$ матрицасы транзитивті болса, онда ол S жиынын қиылыспайтын жиындарға бөледі. 1-типтегі кластеризациялау ақпараттық матрицаға сәйкес келетінін көруімізге болады, оның әрбір жолы тек бірліктерден, ал қалғандары нөлдерден тұрады. Ары қарай топтық шешімдер есебін шешуші құрылымдық қалыпқа келтіруші әдістерінің қалыпқа келтіру алгоритмдерін қарастырамыз. Өйткені транзитивтілік қатынасын қанағаттандырушы құрылымдық түрге топтық шешім матрицасын келтіре отырып, топтық шешімдер есебінің шешуін табудың негізгі тәсілдерін анықтайтын боламыз.

Анықтама 5. 2-типтегі кластеризациялау алгоритмдері деп $S_1, S_2, S_3, \dots, S_m$ берілген объектілерінің $J_m(S)$ сипаттамаларын $K(\|\alpha_{ij}\|_{m \times l})$, $\alpha_{ij} \in \{0,1\}$, $i=1,2,\dots,m$, $j=1,2, \dots, l$ эквиваленттік класына ауыстыру алгоритмін айтамыз:

$$A(J_m(S)) = K(\|\alpha_{ij}\|_{m \times l})$$

1-типтегі кластеризациялау нәтижелерін компоненттері S объектілеріне сәйкес келуші кластер номерлерін анықтайтын m ұзындықты l – мәнді векторын $H = (\beta_1, \beta_2, \dots, \beta_m)$ түрінде жазу ыңғайлы болады. Мұндағы $H =$

$(\beta_1,\beta_2,\ldots,\beta_m)$ векторын ақпараттық вектор деп атаймыз. Онда 1-типтегі кластеризациялау алгоритмі $J_m(S)$ сипаттамаларын $K(H)$ жиынына, яғни $H=(\beta_1,\beta_2,\ldots,\beta_m)$-дан кластерлер белгілеулерінің орнын ауыстыру жолымен алынған мүмкін болатын ақпараттық векторлар жиынына ауыстыру алгоритмі ретінде анықталады. Ары қарай, $I = \|\alpha_{ij}\|_{m\times l}$ матрицасы мен $H=(\beta_1\ \beta_2,\ldots,\beta_m)$ векторын нақты кластеризациялау деп атаймыз және оны A арқылы белгілейік. Сонымен A кластеризациялау алгоритмі 1-типтегі алгоритм деп аталады (немесе 2-типтегі алгоритм), егер де оның жұмыс жасау нәтижесінде алынған кластеризациялау 1-типтегі кластеризациялау (2-типтегі кластеризациялау) болып табылса.

Тану теориясының қарқынды түрде дамуы барысында тану мен классификациялаудың әртүрлі идеялары, болжаулар және принциптеріне негізделген көптеген алгоритмдер құрыла бастады. Деректерді талдаудың кең таралған түрінің бірі кластеризациялау есебі болып табылады. Бейне тану есебінің нәтижелік шешімдерін алгоритмдер тобымен табу тиімді шешім алуға мүмкіндік береді. Кластеризациялау алгоритмдері эквивалентті ақпараттық матрицалар жиынына үйретуші деректердің бейнеленуі ретінде анықталған. Кластеризациялау есебін шешу бірмәнді емес, себебі кластеризациялау ең жақсы деп атап көрсететіндей анық берілген сапа критериі жоқ, бірақ сапалы кластеризациялау жүргізетін эвристикалық критерилер мен алгоритмдер саны көп екені белгілі. Кластерлеу алгоритмдері жұмыс жасауда әртүрлі параметрлеріне, деректер құрылымына және т.б. байланысты нәтижелер де әртүрлі болады. Бір деректер жиынына бірнеше алгоритм қолдануда сан түрлі шешімдер аламыз. Осы ерекшеліктерді ескере отырып кластерлік талдау мен тану есептерінде топтық шешімдерді қолдану тәсілі ұсынылған. Шешімдерді топтық қабылдау, атап айтқанда топтық тануды классификаторлар жиынының шешімдерін бірігіп қолдану деп түсіну керек. Топтық шешімдерді зерттеу 1970 жылдардың ортасынан басталған, соңғы кездерде күрделі, үлкен масштабты қолданбалы есептерде кеңінен қолданысқа ие бола бастады.

Көптеген жылдар бойы танымал тану алгоритмдері мен моделдерін зерттеу негізінде тану алгоритмдерінің жалпы теориясы жасалынды. Тану алгоритмдеріне формалды анықтамалары беріліп келді. Оларға алгебралық операциялар енгізіліп, ғылыми жұмыстарда қолданылды.

Топтық шешімдер алгоритмдерінің негізгі мақсаты топтық шешімдер құрушы алгоритмдерге жататын әрбір алгоритмдер арқылы қарастырылып отырған объектілер жиынын жеке жиындарға бөлу. Алынған әрбір жеке нәтижелерді топтастыра отырып, тиімді қорытынды бөлулерді алу болып табылады.

Кластерлік талдаудың әртүрлі әдістерінің және оңтайландыру тәсілдерінің нұсқаларының болуы оларды келісілген (немесе ұжымдық, комитеттік, ансамбльдік) шешімді қалыптастыру үшін бірлесіп пайдалану мүмкіндігін негіздейді. Мұндай шешімді әзірлеу кезінде "әртүрлі көзқарастан" топтастыру жүргізіледі (бұл "көзқарас" тек қарама-қайшылық деген емес, бір-

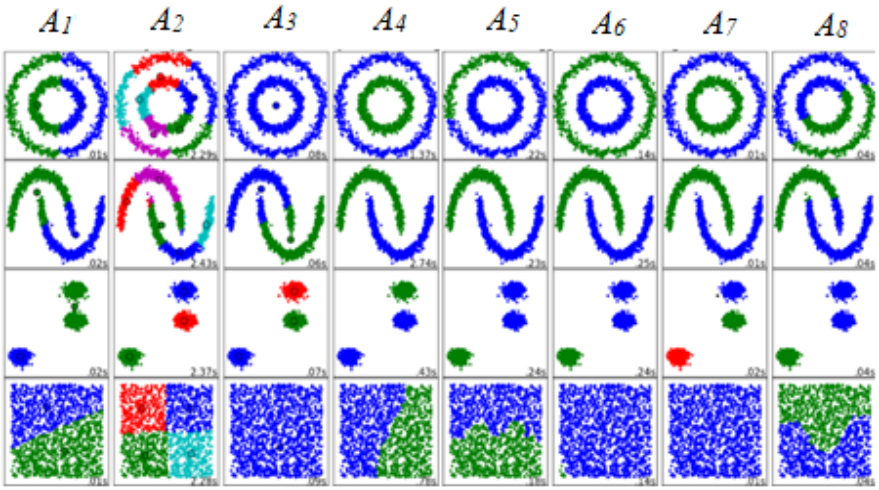
бірін толықтыру деп болжанып отыр), бір нұсқасы басқа нұсқалардың "әлсіз жақтарын" толықтырады). Бұл ретте кластерлер қалыптасатын тұрақты заңдылықтар өзара "күшейтіледі", ал тұрақсыз, керісінше "әлсірейді" [37].

Топтық шешімдер алгоритмдері бастапқы берілгендер жиынын жеке жиындарға бөлулерді келесі төменде көрсетілген жолдармен алуға болады: Кластеризациялаудың әртүрлі алгоритмдерінің нәтижелері; Кластеризациялаудың бір алгоритмін әртүрлі бастапқы баптауларын немесе параметрлерін таңдай отырып бірнеше рет орындау арқылы алынған нәтижелері.

Кластерлік талдаудың топтық шешімдерін құрудың бірнеше негізгі тұжырымдарға негізделеді. Олардың бірі қорытынды кластарға бөлулерді бастапқы берілген объектілер жиынының бірдей немесе басқа кластарға тиесілі екендігін сипаттайтын ақпараттық матрица түрінде көрсету, ал екіншісі бастапқы объектілердегі әр объектілер жұптарының бір немесе әртүрлі кластарға кіретіндігін көрсететін матрица түрінде. Топтық шешімдерді құрудың бұл екінші түрінде объектілерді кластерлерге бөлулердің келісілген матрицасы қолданылады. Бұл матрица коассоциативті матрица деп аталады. Қазіргі таңда коассоциативті матрицаға негізделген топтық кластерлік талдау әдісі де басқа топтық кластерлеу әдістері сияқты кең қолданысқа ие. Коассоциативті матрица берілген бастапқы объектілер жұбы бөлудің әртүрлі нұсқаларында әртүрлі кластерлерде қаншалықты жиі жататындықтарын анықтайды. Сонымен қатар коассоциация матрицасы негізінде объектілер арасындағы ара қашықтықты есептеуге болады. Осы ара қашықтықты қолдана отырып, кластерлік талдауда топтық шешімдер табу есебінде кластерлік талдау алгоритмдерінің көптеген санын қолдануға болады.

Орташаланған матрица элементі ретінде объектілер арасындағы жұптық арақашықтықтар қарастырылуы мүмкін: элемент мәні жоғары болған сайын, сәйкес жұптар әртүрлі кластерлерде болуы жиірек. Қорытынды келісілген кластерлік бөлуді алу үшін кез-келген жұптық ара қашықтық матрицасына негізделген кластерлік талдаудың кез-келген алгоритмін, мысалы, иерархиялық топтауды құрудың агломеративті алгоритмін (дендограммалар құру) қолдануға болады. Топтық шешімде алгоритмнің салмағын анықтау әртүрлі әдіспен жүреді. Топтық шешімді құруда [38]-жұмыста алгоритм салмағын анықтау үшін кластеризациялаудың сапа индексін ескеру ұсынылады.

Топтық шешімдер алгоритмдері нәтижелерінің сапа көрсеткіші қазіргі уақытта қарапайым алгоритмдердің сапа көрсеткішінен арту үстінде, қолданыс аясы кеңейе түсуде. Топтық шешімдер алудың бірнеше тұжырымдамалары бар: коассоцияциялық матрицаны қолданумен, орталық объектілерді ерекшелеу, дауыс беру әдісі т.б. болып табылады.



Сурет 2.2 – A_1-A_8 алгоритмдердің кластерлерге бөлу нәтижелері

	R^1 бөлуі	R^2 бөлуі	...	R^r бөлуі
K_1	1	2		1
K_2	4	4		4
K_3	2	2		1
...				
K_n	3	2		3

Сурет 2.3 – Әртүрлі кластерлерге бөлудің нәтижесі

Берілген 2.2-суреттен A_1-ден A_8-ге дейінгі алгоримдердің бір деректер жиынына қолдануда әртүрлі нәтиже көрсетіп тұрғандығын және 2.3-суретте алгоритмдердің бірдей жиындарға қолданудағы нәтижелерін кесте түрінде көруге болады. Әр алгоритмнің өзінің қолдану аясы бар, қандай да бір деректердің типі әртүрлі болған жағдайда, кластерлерді бөліп алуда бір ғана алгоритм емес әртүрлі алгоритмдердің тобы қолданылады. Топтық шешімдер алгоритмі топтау нәтижелерінің алгоритмдердің параметрлерін таңдаудан тәуелділікті төмендетеді, шулы деректерде сонымен қатар «бос» деректерде орнықты шешім алуға мүмкіндік береді.

2.2 Топтық шешімдер табу есебінің жалпы қойылымы

Айталық $S = \{S_1, S_2, \ldots, S_m\}$ объектілер жиыны берілсін. Мұндағы әрбір $S_i \epsilon S$ объектісі $I(S_i) = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in})$ – белгілемелер жиынтығымен сипатталады, сонымен қатар $A = (A_1, A_2, \ldots, A_t)$ – базалық алгоритмдер жиыны анықталсын, яғни $A_u \epsilon A$, $u=1,\ldots,t$. А базалық алгоритмдер жиынының әрбір алгоритмін алғашқы объектілер жиынына қолдана отырып бос емес K кластерлерден тұратын R_u *шешімін* аламыз:

$$A_u(S, I(S_i)) = R_u; \qquad R_u = K_1 \cup K_2 \cup \ldots \cup K_j;$$

мұнда $K_i \cap K_j = \emptyset$, $i \neq j$, $K_i \neq \emptyset$, $K_j \neq \emptyset$, егер $i = 1,2,\ldots,l$; $j = 1,2,\ldots,l$; $u = 1,2,\ldots,t$.

Әрбір $A_u \epsilon A$ алгоритмдерінің нәтижелерін үйлестіруші R^* нәтижелік бөлуді алу есебін (топтық шешім табу) қарастырамыз. Яғни R_u классификациясының нәтижесі өзара қиылыспайтын, бос емес кластерлер (топтар) жиынынан тұрады. Жалпы алғанда,

$$R^* = \cap R_u;\ u=1,\ldots,t. \tag{2.1}$$

Дегенменде нақты бөлулерді құру мүмкін болатын көптеген нұсқаларды қарастыруды қажет ететін күрделі мәселе болып табылады. Сондықтан қарастырылатын өзекті мәселе топтық шешімдер алгоритмдерінің базалық жинағында бөлулер жиынынан *тиімді нәтижелік бөлуін құру, яғни* берілген функцианалдың экстремалды мәніне негізделген тиімді нәтиже құру болып табылады.

Айталық $\acute{\boldsymbol{K}}$*-классификациялар кеңістігінде* классификациялаулар арасында $d(\boldsymbol{K}, \boldsymbol{K}_i)$ метрикасы анықталсын және $\varphi(\boldsymbol{K}) = \sum_{i=1}^{t} d(\boldsymbol{K}, \boldsymbol{K_i})$, $\boldsymbol{K_i} \subset \acute{\boldsymbol{K}}$, $\boldsymbol{K} \subset \acute{\boldsymbol{K}}$, $i=1,2,\ldots t$. Онда топтық шешім есебі: $\varphi(\boldsymbol{K})$ функционалының минимум мәнін беретін $\varphi(\boldsymbol{K}^*)=\min \varphi(\boldsymbol{K})$, мұнда $\boldsymbol{K} \subset \acute{\boldsymbol{K}}$, $\boldsymbol{K}^* \subset \acute{\boldsymbol{K}}$ S жиыны үшін (2.1) аясында $\boldsymbol{K}^*(S) \subset \acute{\boldsymbol{K}}$ нәтижесін табу. Функционалдардың түрі әртүрлі болуы мүмкін. Диссертациялық жұмыста біз мәні минимумға ұмтылатын функционалды қолдандық, функционалдың экстремалды мәнін қанағаттандыратын тиімді топтық нәтижелік бөлуін алуды, барлық классификациялауға ең жақын классификациялауды табу қарастырдық.

Кластерлік топтық шешімдерді қалыптастырудың негізгі бірнеше тұжырымдары бар. Диссертациялық жұмыста екі тұырым қолданылды. Біріншісі бөлулерді ақпараттық матрица түрінде көрсету, ал екіншісі әр объектілер жұбының бір немесе әртүрлі класқа жататындығын көрсетуші жақындық матрицасы түрінде болады.

Бірінші тұжырым бойынша $I = \left\|\alpha_{ij}\right\|_{m\times l}$ ақпараттық матрицасына келесідей шешуші ереже қолданылады:

$$\alpha_{ij} = \begin{cases} 1, \text{егер } I > \theta, \\ 0, \text{егер } I \leq \theta, \end{cases}$$

мұндағы, θ – арнайы таңдап алынған шек.

Бұл жерде нәтижелік бөлуді мұндай түрде көрсету кластарды қалыптастыру ретіне байланысты айтарлықтай қиындықтарды тудыратындығын айта кету керек. Бұл қажетті нәтижелік бөлуді құруда бір қатар шектеулер туындатады.

Көрсетілген әдістен қарағанда жақындық матрицасы $\left\|\mathrm{T}_{ij}\right\|_{m\times m}$ түрінде әдісті қолданған ыңғайлы болады. Мұнда, нәтиже топтық шешім құрушы

әрбір алгоритмнің нәтижелері арқылы алынады. Әдісті толығырақ қарастырайық. Ол үшін бізге $\left\|M_{ij}^{k}\right\|_{mxm}$, $k=1,\ldots,n$-екінші типтегі кластеризациялау алгоритмдерінің жақындық матрицасының жиыны болсын. Онда $\left\|\mathrm{T}_{ij}\right\|_{mxm}$ келесі түрде алуға болады:

$$\left\|\mathrm{T}_{ij}\right\|_{mxm} = \begin{cases} 1, \text{егер} \sum_{k=1}^{n} m_{ij}^{k} \geq \theta, \\ 0, \quad \text{басқа жағдайларда } 1 \leq \theta \leq n, i \neq j. \end{cases}$$

Бұл жерде T_{ij} матрицасын *топтық шешімдер матрицасы* деп атаймыз. Ал матрица құрушы алгоритмдерді топтық шешімдер матрицасын құрушы деп атайды. Бұл матрица соңғы нәтижені бермейді. Дегенмен де кейбір нәтижелік шешімдер арқылы негізгі есептің шешіміне жуықтаймыз. Ары қарай есеп шешімін негізгі тәсілдер, классификациялау мен бейне танудың алгоритмдерін қолдану арқылы шешуге болады.

Топтық шешімдерді құруды келесі тәсілдер арқылы да құруға болады: орталық объектілерді оқшаулауға негізделген, ядроларды бөліп ала отырып жинақы ішкі жиындарды құру, шоғырлар түріндегі алгоритмдер жиынтығының топтық шешімдер матрицасы, сонымен қатар орталанған жұптық айырмашылықтар матрицасы, графтық тәсілдерді қолдану арқылы алуға болады. Жұмыста топтық шешімдерді құрудың бірнеше әдістері қарастырылып, олар жаңа топтық шешімдер алгоритмдерін құруда қолданылды.

2.3 Жартылай бақыланатын оқыту арқылы классификациялау

Классификациялау есебінде қандайда бір белгілермен сипатталатын объектілерге классификациялау жүргізу қажет болады. Сонымен қатар тану сапасының нақты бір анықталған критериінің тиімді мәнін (мысалы, болуы мүмкін қателіктер бағасын минимизациялау) алу қажет. Классификатор, есептің негізгі қойылымында класс туралы ақпарат барлық объектілері үшін белгілі (бақыланатын оқыту, *supervised learning*) болған жағдайдағы, объектілер-прецеденттерден тұратын оқыту деректері негізінде қалыптастырылады. Бұл жұмыста бейнені тану есебінің қойылымының – жартылай бақыланатын оқыту жағдайындағы классификациялау қарастырылады.

2.3.1 Жартылай бақыланатын оқыту жағдайындағы топтық шешімдер алгоритмдері мен аз рангілі матрица декомпозициясын қолдану

Берілген жұмыста бейнені тану есебін қоюдың бір нұсқасы – бөліктеп басқарылатын классификациялау есебі (semi-supervised classification) қарастырылады. Бұл есепте басапқы деректер объектілерінің бөлігі үшін кластар белгісі белгілі; бар белгіленбеген объектілерді классификациялау

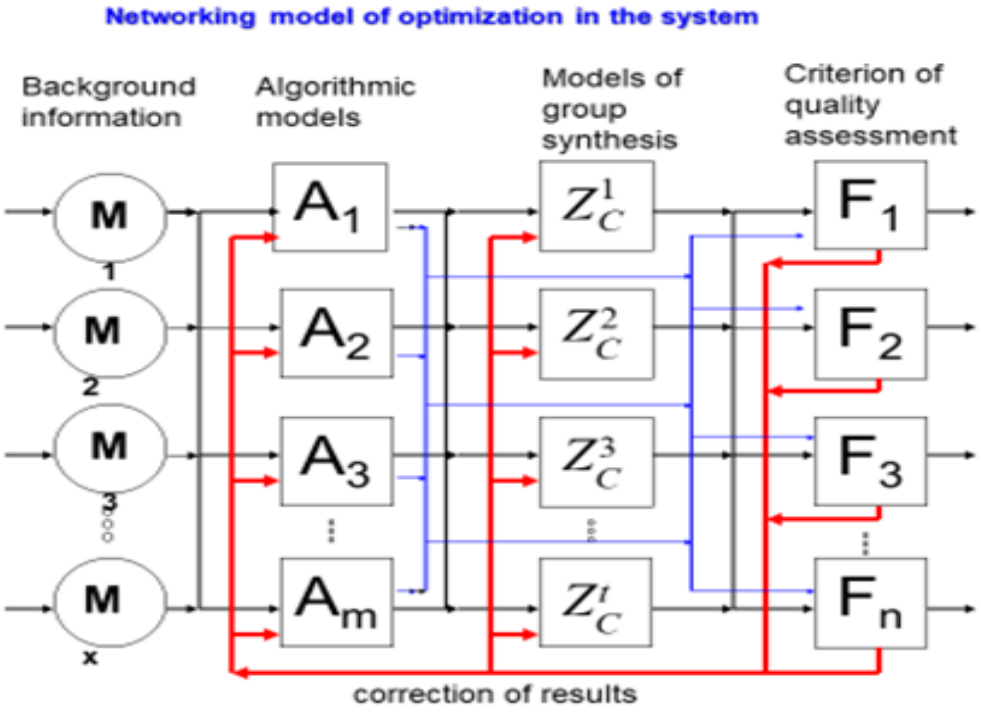
керек немесе жаңа объектілерді тану үшін шешуші ережені қалыптастыру керек. Берілген есеп келесі себептерімен актуалды болып саналады:

- белгіленбеген деректерді қолдануда «шығын аз» (объектілер жататын кластарды анықтауда шығын болады);
- белгіленбеген деректерді белгіленген деректермен бірге қолдану оқыту саасының айтарлықтай өсуін қамтамасыз ететін қосымша ақпаратты алуға мүмкіндік береді.

Жартылай бақыланатын оқыту арқылы классификациялау есебін шешу алгоритмдері өте көп, солардың бірі өзін-өзі оқытудың эвристикалық алгоритмдері, классификациялаудың ядролық әдістері, соның ішінде тірек векторларының және теоретикалық-граф алгоритмдері [39-42] кеңінен қолданылуда.

Деректер ақпараттық емес, шулы белгілер, деректердің құрылымы күрделі немесе кластерлердің нақты саны белгісіз болған жағдайларда алгоритмдердің топтық шешімдерін пайдалану кластерлік талдаудың нәтижелерінің тұрақтылығын арттыруға мүмкіндік береді.

Концептуальды түрде алгоритмдердің топтық шешімдерін (кластерлік ансамбль) қолданудың оптимизацияланған моделын келесі схемадан көруге болады. Осындай модельдер [43] жұмысында ұсынылған.



Сурет 2.4 – Топтық шешімдер нәтижелерін тиімді түрге келтірудің концептуальды схемасы

Схемадағы $M_1,M_2,...,M_\alpha$ – бастапқы кіріс деректері, $A=\{A_1,A_2,\ldots,A_M\}$ – классификациялау алгоритмдері, m-алгоритмдер саны. $z_c^1, z_c^2,\ldots, z_c^t$ – топтық шешімдер алгоритмдері. z_c-дан алынған екінші деңгейі ретінде

алгоритмдердің сапа критериі (дәлдігі) белгіленген. Берілген модель A және z_c жиындарының өлшемін масштабтауға мүмкіндік береді. Модельдегі сызықтардың бағыты өңдеуді қажет ететін деректер ағынын көрсетеді. Суреттен көріп отырғанымыздай тиімділік моделі желілік модель ретінде көрсетілген. Осы модельден жақсы нәтиже алған кезде, нақтырақ айтсақ сапа функционалы экстремалды мәнге тең болғанда шығуға болады.

К алгоритмдерінің тәжірибелік түрде қолдану түрінің бірі [44] жұмыста қарастырылған. Мұнда адам қозғалысын, атап айтқанда бишінің басының қимылдарын тану болып табылатын классификациялауды орындау үшін шартты сызықты гаусианды есептеуде күтуді максимизациялау алгоритмі мен ағаш түріндегі байестік желісі қолданды.

Классификациялау және тану есептерін шешу үшін [45, 46]-жұмыстарда Ю.И. Журавлевтың нейрондық желілерді құруға арналған тәсіл ұсынылған. Тәсіл операторлар теориясына негізделген. Бұл желінің айрықша ерекшелігі ішкі және сыртқы циклдерде аралық есептеулерді айтарлықтай жеңілдететін ішкі қабаттарда активациялаудың диагоналдық функциясын қолдану болып табылады. Сонымен қатар бейнелерді тану үшін CNN нейрондық желісі қолданылған ұқсас жұмыстар [47]-жұмыста сипатталған. Зерттеуде ортаның сәйкес өзгеруін, фильтрлеуді активациялауды ескерусіз, төменірек өлшемді бейне дискрепторын құра отырып аралық қабаттың толықтай шығуы қарастырылады. Осылайша, объектіні визуалды түрде тану сенімділігіне қол жеткіземіз. Кластерлік талдау және алгоритмдер ансамблі қызықты практикалық қолданулары [48, 49]-жұмыстарда сипатталған. Жұмыста деректерді интерпретациялау, урандық кен орындарына арналған геофизикалық каротаждың деректері негізінде стратиграфия және литология шекараларын классификациялау есептері шешіледі. Бірінші жұмыста берілген есеп бірнеше машиналық оқыту алгоритмдері: кездейсоқ орман, логистикалық регрессия, градиентті арттыру, *k*-жақын көршілер және XGBoost әдістері көмегімен шешіледі. Екінші жұмыста жасанды нейрондық желілер *(ANN),* сызықты дискриминалды талдау классификаторы (*LDAC),* тірек векторларын классификациялау *(SVM), k-Nearest-Neighbor (k-NN)* сияқты алгоритмдер қолданылды.

Бұл әдістердің жартылай бақыланатын классификациялауда болатын негізгі шектеулері келесі:

- күрделі есептеулердің күрделілігі және үлкен еске сақтау орнының қажеттілігі;

- шуға тұрақсыздығы.

Жартылай бақыланушы классификациялау есебі [50]-жұмыста кластеризациялау ансамблін қолданумен шешілген. Сонымен қатар мұнда нәтижелер әртүрлі сызбалармен көрсетілген. Көптеген жағдайларда айырмашылықтар матрицасының орнына орташаланған коассоциялық матрицасын қолдану шешімдердің сапасын айтарлықтай арттыратындығы эксперименталдық нәтижелермен дәлелденген. Мұндай дәлелдер теоретикалық талдаулар арқылы [51, 52]-жұмыста көрсетілді: [51]-жұмыста

кейбір тұрақтылық жағдайларында ансамбль өлшемінің артуымен классификациялау қателігі кеми түсетіндігі дәлелденді.

Ұқсастық графының лапласианын (сонымен қатар жиындарды ретке келтіруде танымал) тұрақты түрге келтіруі қолданылатын теоретикалық-графтық тәсілдерде егер деректердің екі нүктесі бірдей жиында жататын болса, онда олар кластардың сәйкес белгілерінен тұратындығының ықтималдығы жоғары болатындығы көрсетіледі. Лапласиан графы белгіленген және белгіленбеген [53] деректерден тұратын жиындарда классификациялаудың тегістік дәрежесін өлшеу үшін қолданылады.

Берілген диссертациялық жұмыста теоретикалық-графтық тәсілдер және кластерлік ансамбль комбинацияларын қолданумен жартылай бақылау арқылы оқыту есебін классификациялаудың жаңа әдісі ұсынылады. Әдістегі негізгі ерекшелігі есептеу шығындарын азайту, объектілерді сақтауда көлемді азайту үшін орташаланған коассоциациялық матрицаны азрангілі түрге келтіре отырып қолдану. Ары қарай жұмыста есептің математикалық қойылымы, бөліктеп бақылау классификациялау әдістеріне қысқаша шолулар жүргізіледі. Әдіске эксперименталдық зерттеулер жүргізілді.

2.3.2 Жартылай бақылау арқылы оқыту есебінің математикалық қойылымы.

Айталық объектілерді танудың басты $Г$ жиынтығы, $K = \{K_1, \ldots, K_k, \ldots, K_K\}$ кластары белгілерінің жиыны берілсін. Әрбір $a \in Г$ объектісі $X = (X_1, \ldots, X_d)$ белгілер жиынымен сипатталады. $S_j(a)$ арқылы a объектісінің S_j белгісінің мәнін белгілейік. Белгілер қабылдауы мүмкін мәндер жиынына, сонымен қатар осы мәндермен жасалынатын операцияларға байланысты белгілер келесі типтерге бөлінеді:

- бинарлы белгі: $S_j(a) \in \{0,1\}$;
- сандық белгі: $S_j(a) \in R$;
- категориялық ерекшелігі: $S(a) \in G_j$ – реттелмеген мәндердің ақырлы жиыны;
- реттік белгі: $S_j(a) \in D_j$ – ақырлы реттелген жиын.

Берілген жұмыста тек қана сандық белгілерді қарастырамыз. Берілген белгілерде $x(a) = (X_1(a), \ldots, X_d(a))$ жиыны $a \in Г$ объектісінің белгілік (бақыланатын) сипатттамасы деп аталады.

Айталық $a_1, \ldots, a_n$ объектілерін бақылайтын $S = \{S_1, \ldots, S_n\}$ деректері берілсін, мұндағы $S_i = S(a_i)$. Бөліктеп бақыланушы оқыту есебіндегі деректерде бақылаудың екі типі бар:

- $S_1 = \{S_1, \ldots, S_{n_1}\}$-кластардың танымал белгілерімен $a_1, \ldots, a_{n_1}$ объектілерін сипаттау;
- $Y_1 = \{y_1, \ldots, y_{n_1}\}$, мұндағы $y_i \in K$ – a_i, $i=1,\ldots,n_1$ объектісі тиісті кластың белгісі;
- $S_0 = \{S_{n_1+1}, \ldots, S_n\}$-белгіленбеген объектілердің сипатталуы.

Есептің қойылымының бірінші нұсқасында индуктивті оқыту, яғни класс номерлерін S_0 - объектілері ретінде де, кездейсоқ жаңа бақылауларына да сәйкес қоятын классификаторларды құрастыру. Есептің қойылымының екінші нұсқасында трансдуктивті оқытуды жүргізу, яғни $Y_0 = \{y_{n_1+1}, \dots, y_n\}$ кластар белгісін тек S_0 -ден алынған объектілері үшін анықтау қажет етіледі. Бірінші және екінші жағдайларда классификациялаудың кейбір сапа функциоалы қолданылады. Берілген жұмыста есеп қойылымының екінші нұсқасы қарастырылады.

2.3.3. Жартылай бақылау арқылы оқыту әдісі

Жартылай бақыланатын оқытудың кейбір жиі қолданылатын тәсіліндерін қарастырайық.

Өзін-өзі оқыту әдісі: Берілген тәсілде оқытушы арқылы классификациялаудың бірнеше базалық алгоритмі қолданылады. Алгоритм бірінші қадамда белгіленген деректермен оқытылады, содан кейін белгіленбеген бөлігіне классификациялау жүргізіледі. Сонымен қатар әр классификацияланушы объектіге танудың сапасының бағасы есептелінеді. Келесі қадамда қандай да бір берілген шектен жоғары сапа бағасының бақылауы X_0 жиынынан алынып тасталады да X_1-ге қосылады, ал олардың белгілерін Y_1-ге толтырылып, ары қарай базалық алгоритм белгіленген деректермен оқытылу үшін және қалған белгіленбеген бөлігін тану үшін тағы да қолданылып, белгіленбеген объектілер қалмағанша қайталанады.

Тірек векторларының трансдуктивті әдістері

Берілген типтегі әдіске тірек векторларының (SVM) әдісі жатады. Есептің базалық қойылымында (бинарлы классификациялау есебінде) жолақтың бөлуші класының максималды енін, бөлуші гипержазықтықтың бағытын табу қажет етіледі. Алгоритмдердің кірісіне $Y=\{y_1,\dots,y_l\}$, $i=1,\dots,n$ кезінде $y_i \in \{-1, +1\}$ кластар белгілерімен X объектілерінің оқытатын деректері беріледі. Кластардың сызықты бөлінулері жағдайында бөлуші гипержазықтықтардың саны шексіз болады. Кластардың екеуіне де ара қашықтықтары максималды болатындай гипер жазықтық таңдап алуға болады. Сондықтан бөлуші жолақ шетінде жатқан нүктелер тірек векторлары деп аталады.

Гипержазықтық теңдеуін $\langle w, x\rangle + b = 0$ түрінде көрсетейік, мұндағы $\langle \cdot, \cdot \rangle$ - скалярлы көбейтінді, w - бөлуші гипержазықтыққа перпендикуляр вектор, ал b - көмекші параметр. Тірек векторларының әдісі келесі түрдегі шешуші функцияны құрады

$$f(x)=sign(\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x\rangle + b)$$

мұндағы $\alpha_1,\dots,\alpha_n \geq 0$ – кейбір параметрлер; сонымен қатар тірек векторлары үшін келесі шарт орындалуы үшін гипержазықтық коэффициенттерін қалыпты түрге келтіру жүргізіледі:

$$\langle w, x_i\rangle + b = \pm 1$$

Бұл жерде атап айта кету керек қосындылау $\alpha_i \neq 0$ болатындай тек тірек векторлары бойынша жүреді. Тірек векторларының трансдуктивті әдісінде гипержазықтықты тек X_1-ден алынған белгіленген нүктелер ғана емес, X_0 -ден алынған белгіленбеген нүктелерді максималды алшақтықпен бөліп тұратындай түрде жүргізу керек. Осылайша, гипержазықтық максималды төмен тығыздықтағы аймақта болуы қажет. Тиімді гипержазықтықты іздеудің тиімділік есебін келесідей түрде құрастыруға болады:

$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, i = 1, \dots, n, \ \xi_i \geq 0, i=1,\dots,n \text{ шектеуінде}$$

$$Y_0, w, b, \xi : \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \rightarrow \min_{Y_0,w,b,\xi}$$

табу керек. Мұндағы $\xi_1,\dots, \xi_n$ - жолақтың шекарасының нүктелеріндегі ауытқуларға қойылатын айыппұл айнымалылары. Бөлуші жолақтың енінің ең үлкен мәнін максимизациялау жүргізіледі.

Берілген есепті жуықтап шешу алгоритмі бар; ол үшін $\alpha_1,\dots,\alpha_n$ параметрлеріне сәйкес қосарлы есебі шешіледі.

Кластардың сызықты бөлінуін X бастапқы кеңістігін X' үлкен өлшемді жаңа кеңістігіне $\varphi: X \rightarrow X'$ түрлендіруі жүзеге асыра алады. Осы $\langle x, z\rangle$ скалярлық көбейтінділердің арқасында X' кеңістігінде $\langle \varphi(x), \varphi(z)\rangle$ түріндегі көбейтінділермен алмастыруға болады. Мұндай жағдайда шешуші функция келесі түрде болады:

$$f(x)=sign(\sum_{i=1}^{n} \alpha_i y_i \langle \varphi(x_i), \varphi(x)\rangle + b).$$

$K(x,z)=\langle \varphi(x), \varphi(z)\rangle$ функциясы ядро деп аталады. Ядроны таңдау түзетуші кеңістікке өтуді анықтайды.

Мерсер теоремасы: Үзіліссіз жиында берілген $K(x,z)$ функциясы ядро болып табылады, егер ол симметриялы $K(x,z)=K(z,x)$, теріс емес анықталған болса, $\forall z \in R^p$, $z^T Kz$ [54].

Тірек векторларының трансдуктивті әдісі үшін тиімділік есебі дөңес емес, ал оның жуықтаған шешімдерінің танымал алгоритмдері бақылау сандарынан тәуелді полиноминалды күрделікте болады. Сондықтан берілген тәсіл салыстырмалы түрде үлкен емес көлемдегі деректерге қолдануға болады(мыңға жуық бақылаулар).

Ұқсастық графының лапласианын ретке келтіру

V төбелер жиынында X-тан алынған бақылаулармен сәйкес келетін, ал E қабырғалар жиыны (x_i,x_j), $i,j=1,\dots,n$, $i \neq j$ жұбына жауап беретіндей $G=(V,E)$ салмақты бағытталмаған толық графын қарастырайық. Әрбір (x_i,x_j) қабырғаларына берілген нүктелер жұбының ұқсастық дәрежелерінің мағынасынан тұратын W_{ij} (салмағы) теріс емес саны қойылған. Мысалы, салмақты ұқсастықтың радиалды базистік (гаустық) функциясы көмегімен анықтауға болады:

$$W_{ij}=exp\left(-\frac{\|x_i-x_j\|}{2\sigma^2}\right)$$

мұндағы, σ –берілген параметр.

Айталық $Y_i=(Y_{i1},\dots,Y_{iK})$ $Y_{ik}=I[y_i=c_k]$ кластарына тиістілігін бақылайтын бульдік векторды белгілейді, мұндағы I[·] -предикатты функция: I[*true*]=1, I[*false*]=0, $i=1,\dots,n_1$, $k=1,\dots,K$. $F_i=(F_{i1},\dots,F_{iK})$ арқылы $F_i \geq 0$ элементі c_j класына x_i нүктесінің тиістілігінің бағалану дәрежесін білдіретін классификацияланушы векторды белгілейік және $n\times K$ өлшемдегі классификацияланушы матрицасы $F=(F_1,\dots,F_n)^T$ ретінде анықталған болсын.

Келесі тиімділік есебін қарастырайық: $F \geq 0$ шартында келесі теңдеуді табу керек, мұндағы $\beta > 0$ -реттеуіш параметрі.

$$F^* = \arg\min_{F\in R^{n\times K}} Q(F) =$$

$$=\frac{1}{2}\left(\sum_{x_i\in X_1}\|F_i-Y_i\|^2+\beta\sum_{x_ix_j\epsilon X}W_{ij}\left\|\frac{F_i}{\sqrt{D_{ii}}}-\frac{F_j}{\sqrt{D_{jj}}}\right\|^2\right), \qquad (2.1)$$

Бұл теңдеудің оң жақ бөлігіндегі алғашқы қосындысы белгіленген деректерді есептеуде қателіктерді минимизациялауға арналған; екінші компоненті деңгейлестіру функциясының ролін атқарады: оны минимизациялауда егер екі x_i, x_j нүктелері (белгіленбеген де, белгіленген де болуы мүмкін) ұқсас болса, онда олардың классификацияланушы векторларының арасында айтарлықтай айырмашылықтары болмауы керек. Тиімділік функционалы дөңес болып табылатыны белгілі.

Элементтері $D_{ii}=\sum_j W_{ij}$ болатын D диагоналдық матрицасын қарастырайық. $L=I-D^{-1/2}WD^{-1/2}$ матрицасын қалыпқа келтірілген лапласиан деп атайды, мұндағы I - бірлік матрица. Матрицаның өлшемі $n\times n$, оның L_{ij} элементтері $L_{ij}=\delta_{ij}-\frac{W_{ij}}{\sqrt{D_{ii}}\sqrt{D_{jj}}}$, мұндағы $\delta_{ij=}I[i=j]$-Кронокер символы. Тиімді шешімдерді табу үшін (2.1)-теңдеуді дифференциалдайық, түрлендіруден кейін алатынымыз:

$$\frac{\partial Q}{\partial F_{ik}}|_{F_{ik}=F_{ik}^*}=F_{ik}^*-Y_{ik}+\beta F_{ik}^*-\beta L_{i,\cdot}F_{\cdot,k}^*=0, \qquad (2.2)$$

$$\frac{\partial Q}{\partial F_{ik}}|_{F_{ik}=F_{ik}^*}=\beta F_{ik}^*-\beta L_{i,\cdot}F_{\cdot,k}^*=0, \qquad (2.3)$$

мұндағы $L_{i,\cdot}, F_{\cdot,k}^*$ - L матрицасының i-ші жолы және сәйкесінше F^* матрицасының k-шы бағаны, $i=1,\dots,n_1$, $k=1,\dots,K$. $n\times K$ өлшемді

$$Y_{1,0}=(Y_1,\dots,Y_{n_1},\underbrace{0,\dots,0}_{n-n_1})^T$$

матрицасын $Y_{1,0}$ арқылы белгілейік, ал $n\times n$ өлшемді

$$I_{1,0} = diag(I_{11} \dots, I_{nn}), I_{ii} = \begin{cases} 1, i = 1, \dots, n_1 \\ 0, i = n_1 + 1, \dots, n, \end{cases}$$

диагоналдық матрицасын $I_{1,0}$ арқылы белгілейік.

Онда (2.2), (2.3) теңдеулерін келесі түрде:

$$(I_{1,0} + \beta L)F^* = Y_{1,0}, \quad (2.4)$$

мұнда $$F^* = (I_{1,0} + \beta L)^{-1}Y_{1,0} \quad (2.5)$$

Алынған теңдеулерден $F = F^*$ матрицасын есептеуден кейін ақырғы классификациялау келесі төменде көрсетілген формуламен анықталады:

$$y_i = c_{k^*}, \text{ мұндағы } k^* = arg \max_{k=1,\dots,K} F_{ik}, i = n_1 + 1, \dots, n.. \quad (2.6)$$

2.3.4 Ұсынылған әдіс

Орташаланған коассоциативті матрицаны алуда $\{P_l\}_{l=1}^r$ бөлу нұсқасын қарастырайық. Әр P_l нұсқасы үшін

$$H_l=(H_l(i,j))_{i,j=1}^n$$

Орташаланған коассоциативті матрица төмендегідей анықталған:

$$H = \left(H(i,j)\right)_{i,j=1}^n, H(i,j) = \sum_{l=1}^r w_l H_l(i,j)$$

мұндағы w_1,…,w_r - алгоритмдердің топтық шешімдері элементтерінің салмағы, $w_l \geq 0$, $\sum w_l = 1$. Салмақтары бірдей немесе кластеризациялаудың әр нұсқаларының кейбір көрсеткіштерінің сапасына пропорционалды таңдап алынады.

Келесі қасиеті матрицалық есептеудің тиімділігін айтарлықтай арттырады.

Тұжырым. Өлшемді орташаланған ассоциация матрицасы

$$H = BB^T, B = [B_1 B_2 \dots B_1], \quad (2.7)$$

түріндегі төменгі дәрежелі *ыдырауға мүмкіндік береді, мұндағы* B *- блоктық матрица,* $B_l = \sqrt{w_l A_l}$, A_l *- l бөлу үшін* $n \times K_l$: $A_l(i,k) = I[c(x_i) = k]$, *i=1,…,n, k=1,…,*K_l*, өлшемдегі ассоциация матрицасы.*

Ереже бойынша, $m = \sum_l K_l = n$ теңдеуі $n \times n$ өлшемді толық орташаланған коассоциация матрицасының орнына $n \times m$ өлшемдегі B матрицасының ыдырауын сақтай отырып, (2.5)-теңдеуі жадыны үнемдеу мүмкіндігін беретіндей орындалады. Матрицаның $H \cdot x$ векторына көбейту күрделілігі $O(n^2)$-ден $O(nm)$-ге дейін төмендейді.

Кластерлік талдау алгоритмдерінің топтық шешімі және ұқсастық графының лапласианымен реттеу.

H матрицасын ұқсастық матрицасы ретінде алып, $\tilde{L} = I - \tilde{D}^{-1/2} H \tilde{D}^{-1/2}$, мұндағы $\tilde{D} = diag(\tilde{D}'_{11}, \dots, \tilde{D}_{nn})$, $\tilde{D}'_{ii} = \sum_j H(i,j)$ түріндегі сәйкес графтың қалыптандырылған лапласианын қолданайық. Онда алатынымыз:

$$\tilde{D}_{ii} = \sum_{j=1}^{n} \sum_{l=1}^{r} w_l \sum_{k=1}^{K_l} A_l(i,k) A_l(j,k) = \sum_{l=1}^{r} w_l \sum_{k=1}^{K_l} A_l \text{ (j, k)} =$$
$$= \sum_{l=1}^{r} w_l \, n_l(i), \qquad (2.8)$$

$\tilde{L}$ -ді (2.4)-формулаға қойып, төмендегі сызықтық теңдеулер жүйесін құрайық:

$$\left(I_{1,0} + \beta L\right) F^{**} = Y_{1,0}$$

Бұл жүйе (2.7)-теңдеуді қолдануда азрангілі матрицасымен тиімдірек түрде жасалынатын операцияларды қолданатын формаға түрлендірілуі мүмкін. Ол үшін $U = \tilde{D}^{-1/2} B$ арқылы белгілейік, онда $\tilde{L} = I - UU^T$ аламыз. (2.5) және (2.7)-ден келесі жүйені аламыз:

$$\left(I_{1,0} + \beta I - UU^T\right) F^{**} = Y_{1,0}$$

Оны сандық шешу үшін жуықталған итеративті алгоритмдердің бірін қолдануға болады. Берілген жұмыста градиенттік түсуге негізделген GDSolve алгоритмі қолданылады. GDSolve алгоритмінің жинақтылығы жүйенің симметриялық оң анықталған жағдайы үшін [55]-жұмыста дәлелденген. Лапласианың азрангілі көрсетілуін қолданушы берілген GDSolveLR алгоритмінің модификациялануының негізгі қадамдарын сипаттайық.

Алгоритм GDSolveLR:

Кіріс деректері: U, $I_{1,0}$: (2.8) жүйенің оң жағындағы сирек толтырылған матрицасы; $Y_{1,0}$: (2.8) жүйенің оң жағындағы матрица; δ - шешімнің қажетті дәлдігі.

Шығыс деректері: F^{**}: классификациялаудың ізделініп отырған матрицасы.

Алгоритм қадамдары

1. t:=0; $F^{**}(0) := 0$;
2. for *k:=1* → *K* do
3. b:= $Y_{1,0 \cdot k}$ ($Y_{1,0}$ матрицасының k -шы бағаны);

4. repeat
5. қарама-қайшылықты есептеу
$r(t) \coloneqq b - (I_{1,0} \cdot F_{\cdot,k}^{**}(t) + \beta F_{\cdot,k}^{**} - \beta U(U^T \cdot F_{\cdot,k}^{**}(t)))$;
6. қадамның тиімді ұзындығын табу

$$\eta(t) \coloneqq \frac{r(t)^T r(t)}{r(t)^T (I_{1,0} \cdot r(t) + \beta r(t) - \beta U(U^T \cdot r(t)))};$$

7. $F_{\cdot,k}^{**}(t+1) \coloneqq F_{\cdot,k}^{**}(t) + \eta(\mathrm{t}) \cdot r(t)$;
8. until r(t)$< \delta$;
9. end for
10. return $F^{**}(t+1)$.

Бұл жерден байқайтынымыз 5, 6-қадамдарда аз рангілі түрде көрсетілген лапласиан матрицасына көбейту жүргізіледі; сонымен қатар $n \times n$ өлшемдегі матрицасын толықтай сақтаудың керегі жоқ.

Сонымен ұқсастық графының лапласианына, коассоциациялық матрицаның азрангілі жіктелуі, градиенттік әдісіне негізделген классификациялаудың бөліктеп бақыланатын SSC-LR-GD алгоритмін сипаттайық.

Алгоритм SSC-LR-GD:

Кіріс деректері:

X: X_1 берілген класты объектілер жиыны және X_0 белгіленбеген объектілер;

Y_1: белгіленген объектілер үшін кластар белгілерінің жиыны; *r*: кластеризациялау алгоритмдерін жүргізу саны;

Ω: кластеризациялау алгоритмдерінің жұмысының параметрлер жиыны; топтаудың сапасын бағалау тәсілі.

Шығыс деректері:

X_0-ден алынған объектілер үшін кластардың алдын ала болжамымен алынған белгілері

Алгоритм қадамдары

1. *X*-тен алынған объектілерді кластеризациялаудың r нұсқасын кластерлік талдау алгоритмінің параметрін Ω-ден кездейсоқ таңдап ала отырып қалыптастырайық; $w_1, \ldots, w_r$ салмақтарының нұсқаларын (топтаудың сапа бағалауын) табу керек;

2. (2.5)-формуладағы *B* матрицасын, (2.8)-дан $\widetilde{D}$-ны және $I_{1,0}$ қолдана отырып қалыптандырылған лапласианды азрангілі түрде есептеу керек.

3. F^{**}классификациялық матрицасын GDSolveLR алгоритмінің көмегімен табу керек;

4. (2.6)-ға $F = F^{**}$ матрицасын қолдана отырып Y_0 үшін белгіні анықтау керек.

end.

2.3.5 Ұсынылған әдіске жүргізілген эксперименттік зерттеулер

Шулы эффекттермен және деректердің әртүрлі көлеміндегі шарттарында алынған алгоритмнің тиімділігін тексеруге қатысты сандық эксперименттер жүргізілді. Алгоритм жұмысының сапасы:

а) берілген үлестіруге бағынышты көп рет генерацияланған жасанды деректерге;

б) гиперспектралды бейнелерді нақты тану есебінде анықталды.

Берілген мысалда бірдей салмақтағы $N(a_i,\sigma_X I)$ көпөлшемді қалыпты үлестірудің бес қоспасынан генерацияланған деректер жиыны қарастырылған; $a_i \in R^d$, $i=1,\ldots,5$, $d=8$; σ_X шамасы берілген параметр болып табылады.

Алгоритмнің тұрақтылығын зерттеуде, шулы деректер кезіндегі $U(0,\sigma_2)$ біркелкі үлестірілуіне қосылатын екі тәуелсіз кездейсоқ шамалар генерацияланады. Шу параметрі $\sigma_2 = 5$. 2.5-суретте генерацияланған деректердің мысалы келтірілген.

Эксперимент жасау кезінде объектілердің 10% берілген деректердің белгіленген бөлігін құрады; қалған нүктелері белгіленбеген деректерге тиесілі.

Топтық шешімдер нұсқалары К-орта топтар алгоритмінде бастапқы центроидтар ретінде нүктелерді кездейсоқ таңдау жолымен құрастырылды (кластерлер саны тура 10-ға тең). Алгоритмдер саны: $r=10$. Ансамбль элементтерінің салмағы бірдей: $w_l \equiv 1/r$ Реттеуіш β параметрі кросвалидация мен 0,005 қадаммен [0.001, 0.5] интервалында тор бойынша іздеуді қолданумен бағаланды; жақсы нәтижелер $\beta = 0.1$ мәнінде алынды. параметрі δ=10^{-5}. Салыстыру үшін, шешімі (2.5) формула негізінде қабылданатын, RBF ядросының көмегімен ($\sigma = 4$ параметрі) бағаланған стандартты ұқсастық матрицасы қолданылған алгоритм қарастырылды.

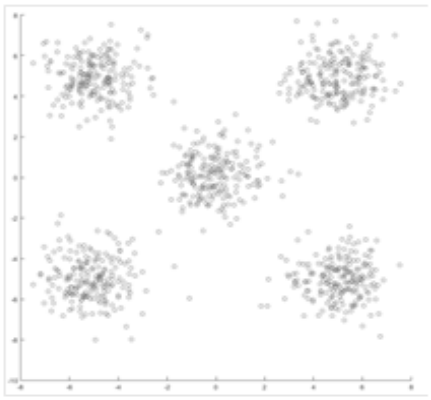
Төмендегі кестеде эксперименттердің нәтижелері көрсетілген. Шешімдер дәлдігінің орташалаған көрсеткішіне қосымша ретінді, кестеде алгоритмдер жұмысының орташаланған уақыты келтірілген (2,8 ГГц және 4 ГБ ОЕСҚ-ты Intel Core i5 екі ядролы процессорында). SSC-LR-GD алгоритмі ретінде топтық шешімдер уақыты мен матрицалық түрлендіруге кеткен уақыт(секундпен) жеке көрсетілген. Қою шрифтпен дәлдік бағалауы ерекшеленген (p -value$<$ 10^{-5}). 2.1-кестеден де көрсетілген нәтижелерден байқайтынымыз, берілген эксперименттерді SSC-LR-GD алгоритмі SSC-RBF алгоритміне қарағанда дәлдігі айтарлықтай жоғары дәрежеде екендігін көрсетті. Деректердің $n = 10^5, n = 10^6$ үлкен көлемі үшін SSC-RBF алгоритмі жадының көлемі (74.5 ГБ и 7450.6 ГБ) жеткіліксіз болғандықтан жұмыс істеуі мүмкін болмады. SSC-LR-GD алгоритмінің жұмыс істеуінің негізгі уақыты ұқсастық матрицасын аз рангілі түрде қолданбайтын SSC-RBF ұқсас матрицасынан қарағанда айтарлықтай аз болды.

Гиперспектральды бейнелерді талдау. Алынған алгоритмді эксперименттік түрде зерттеу үшін 400-2500 nm диапазондағы 224 спектральды каналдардан тұратын, өлшемі 145 те 145 пиксель болатын гиперспектральды Indian Pines [56] бейнесі қолданылды.

Кесте 2.1 – Әртүрлі көлемдегі n-деректер және σ_X параметрі мәнінде үлестіру қоспасымен жасалынған эксперименттер нәтижелері

N	σ_x	SSC-LR-GD			SSC-RBF	
		Дәлдігі	t_{ens} (sec)	t_{matr} (sec)	Дәлдігі	Уақыт (sec)
1000	1	1.000	0.06	0.10	1.000	0.32
	3	0.985	0.07	0.02	0.982	0.32
	5	0.874	0.13	0.11	0.817	0.35
3000	1	1.000	0.10	0.48	1.000	5.42
	3	0.986	0.13	0.10	0.984	5.29
	5	0.878	0.23	0.48	0.848	5.48
10^5	1	1.000	2.05	25.69	-	-
10^6	1	1.000	49	443	-	-

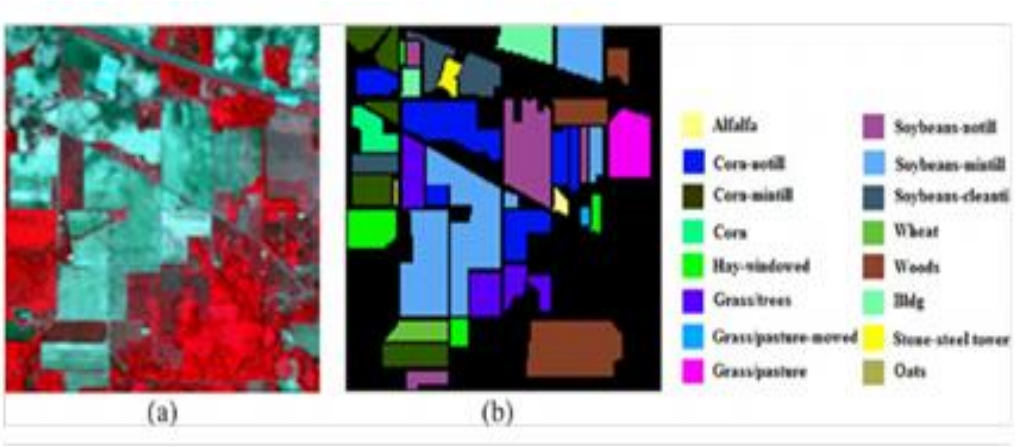
Төменде берілген 2.6 (a)-суретінде RGB-композит бейнесі, ал 2.6 (b)-суретінде бейнені 16 тематикалық кластарға эталондық бөлу келтірілген. Суретте кластардың ешқайсына да жатпайтын белгіленбеген пикселдер бар. Берілген пикселдер талдау кезінде қарастырылмады. Эксперимент жасауда белгіленген бөлігі әр компоненттер үшін кездейсоқ түрде таңдап алынған нүктелердің 1% құрады. Спектральды каналдардың корреляция әсерін азайту үшін бастапқы берілгендерден басты компонент әдісі көмегімен 10 жаңа белгілер алынды.



Сурет 2.5 – Генерацияланған деректер (X_1, X_2 жазықтықтарына проекциясы): n=1000, $\sigma_x = 1$

Құрылған деректер кестесі SSC-LR-GD алгоритмінің кіріс деректері ретінде алынды. Топтық шешім алгоритмдерінің саны *r=10*, ал бөлудің әртүрлі нұсқалары *[1000, 1000+r]* интервалында кластерлердің санын вариацияланған алынды. Қалған параметрлері алдыңғы эксперименттерде сипатталған параметрлермен сәйкес келеді.

SSC-LR-GD алгоритмінің жұмыс істеуінің орташа уақыты 1.5 минут, ал SVM – 0.3 минутты құрады. Қорытындысында SSC-LR-GD орташа дәлдігі 0.557, ал SVM - 0.513. Жұптық *t*-тесті SSC-LR-GD (*p*-value < 10^{-5}). алгоритмінің дәлдігінің айтарлықтай жоғары екендігін көрсетті.



Сурет 2.6 – Генерацияланған деректер

2.4 Шоғырлар түріндегі алгоритмдер жиынтығының топтық шешімдер матрицасы

Топтық шешімдер матрицасының $\left\|\mathrm{T}_{ij}\right\|_{m\times m}$ ұқсастық матрицасы келесідей:

$$\left\|\mathrm{T_{ij}}\right\|_{\mathrm{m\times m}} = \mathrm{J(T_{ij})} = \begin{cases} 1, \text{ егер } \sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{m_{ij}^{k}} \geq \theta, \\ 0, \text{басқа жағдайда } 1 \leq \theta \leq \mathrm{n}, \mathrm{i} \neq \mathrm{j}. \end{cases} \quad (2.9)$$

мұндағы әрбір $\theta > 0$ сандық шектеу және

$$m_{ij} = \begin{cases} 1, \text{ егер } & S_i \in K_k \\ 0, \text{ егер } S_i \in K_k, & S_j \notin K_k \end{cases}$$

Осы (2.9)-теңдеуге $\mathrm{A_1}$, $\mathrm{A_2}$, …, $\mathrm{A_t}$ әрбір алгоритмдері үшін $w_1\ w_2\ ,\ldots,w_t$ салмақтық коэффициенттерін енгізейік. Онда (2.9)-теңдеуді келесі түрде жазуға болады:

$$J(\mathrm{T}_{ij}) = \begin{cases} 1, \text{ егер } \sum_{k=1}^{n} w_k m_{ij}^{k} \geq \theta, \\ 0, \text{басқа жағдайда } 1 \leq \theta \leq n, i \neq j. \end{cases} \quad (2.10)$$

Берілген (2.10)-есебінің негізгі мақсаты топтық шешім матрицасының тиімді құрылымдық қалпын құру болып табылады және оны қалыпқа келтіруші деп атап, V арқылы белгілейік. Сонымен V - (2.10)-шешуші ережесінің қалыпқа келтірушісі. Енді осы топтық шешім матрицасының құрылымындағы ішкі бағалау параметрлерін қарастырайық: $\gamma_1,\gamma_2,\ldots,\gamma_t$. Ал салмақтық коэффициенттер топтық шешімдер алгоритмдерінің сапаларын сырттай бағалайды. θ-шектеуі де $J(\mathrm{T}_{ij})$ шешуші ережесінің негізгі параметрі болып табылады, ол T_{ij}-топтық шешім матрицасының құрылымын анықтайды.

Сонымен T_{ij} топтық шешімдер матрицасының негізгі параметрлері: $\gamma_1,\gamma_2,\ldots,\gamma_t$ топтық шешімдер алгоритмдерінің ішкі бағалау параметрлері, $w_1,w_2,\ldots,w_t$ алгоритмдердің салмақтық коэффициенттері және θ-шектеуі. Бұл аталған параметрлердің барлығы да топтық шешімдер матрицасының құрылымдық қалыбын қалыптастыруда әсері бар. Айталық ішкі бағалау алгоритмдері топ құрушы алгоритмдердің әрқайсысының сапасын анықтайтын болғандықтан олардың мәндерінің өзгеруі топтық шешімдер матрицасының құрылымдық қалпына, яғни алгоритмдер арқылы құрылған кластар тобының құрылымына әсер етеді. Ал салмақты коэффициенттерінің параметрлері алгоритмдерді сырттай бағалауға арналған. Сонымен қатар θ-шектеуі негізгі параметр екендігін атап айтып өттік.

Жалпы топтық шешім матрицасының құрылымын граф, түйіндері объектілер жиыны, ал объектілерінің жалғасуы (қабырға деп аталатын) жұпты екі объект болып келетін тор түрінде бейнеленетін граф түрінде қарастыруға да болады. Сонымен $\left\|T_{ij}\right\|_{mxm}$ топтық шешім матрицасының құрылымын келесі төмендегі шартты қанағаттандыратын $T_G(\left\|T_{ij}\right\|_{mxm})$ графымен алуға болады:

$$T_B = \begin{cases} T_{ij} = 1 \text{ онда бірінші ретті тор қабырғасы,} \\ T_{ij} = 0 \text{ онда екінші ретті тор ққабырғасы.} \end{cases}$$

Яғни топтық шешім матрицасының құрылымдық қалпы ретінде $T_G(T_B,S)$ графын қарастыруға болады. Енді бұл бөлімде осы топтық шешім матрицасын сонымен қатар бұл матрицаға қатысты тиімділік есебін толығырақ қарастырайық.

Алдымен θ-шектеуі шектік мәндерге тең болған жағдайын қарастырамыз: Мұнда $\theta=1$ болған жағдайында, топтық алгоритмдердің бірі объектілер жұбын бір класқа жатқызса онда кез-келген объектілер жұбы бір класқа жатады. Бұл жағдайда байқайтынымыздай топтық шешімдер матрицасының құрылымы бір үлкен кластан тұруы мүмкін. Сонымен қатар онда оқшауланған объектілер де болуы мүмкін.

Ал $\theta=k$ жағдайында әр объект жеке шоғырлар түрінде қарастырылады. Мұнда топтық шешімдер алгоритмдері бір шоғырға жатқызған байланысушы объектілер жұптары қарастырылмайды.

Топтық шешімдер нәтижелерінің тиімділігін арттыруда сапа функционалдары қолданылады. Сапа функционалдарының көптеген түрлері бар. Сол функционалдардың кез-келген түрі қолданылуы мүмкін. Берілген объектілер тобын ақырғы кластерлерге бөлуде «тұрақты» ақырғы бөлулерді алу тәсілі қолданылады. Мұнда $S_Г$ объектісін $К_i$ класының тұрақты объектісі деп қарастырады, егер келесі шарт орындалса: $T_Г(S_Г,К_i)>T_Г(S_Г,К_j)$, мұндағы $T_Г(S_Г,К_i)$ - $S_Г$ объектісінің $К_i$ класына дейінгі жақындығы. Бұл жерде мұндай объектілер санына сапа функционалы арқылы бағалау жүргізіледі.

Берілген объектілерді ақырлы тиімді бөлуді алудың тағы бір жолы салмақтық коэффициенттерге байланысты.

Айталық A_t, $t=1,2,\ldots,n$ алгоритмдер және бастапқы объектілер жиыны S_t берілсін. Берілген алгоритмдерді қолдану нәтижесінде сапа функционалдарының мәндерін есептеп аламыз: $\varphi(P_1), \varphi(P_2),\ldots, \varphi(P_n)$. Мұндағы $P_t=A_t(S)$, $t=1,2,\ldots,n$. Сапа функционалының ішінен максималды мәнін анықтап, сыртқы бағалау парамметрлері w мәндерін есептейік:

$$w = \varphi(P_t)/\varphi', \; t=1,2,\ldots,n$$

мұндағы, φ'-сапа функционалының максималды мәні. Нәтижелік w салмақтық коэффициенті топтық шешімде қолданылушы әрбір алгоритмдердің деңгейін анықтайды, сонымен қалыпқа келтіруші V-ның шешуші ережесінде қолданылады. Сонымен салмақтық коэффициенттер сапа функционалдарының әртүрлі қалыптандырылған немесе реттелген мәндері ретінде алынады.

2.5 Кластардың орталық объектілерін оқшаулауға негізделген топтық шешім алгоритмі

Бейнелерді тану есебі қандай да бір сапа критерилері бойынша объектілер жиынын бірнеше кластарға бөлу. Әдетте оқыту объектілерінің барлықтарының маңыздылықтары (құндылықтары) бірдей бола бермейді. Олардың ішінде *орталық объектілері* болады. Егер классификацияланушы объекті есептеулер нәтижесінде осы орталық объектіге жақын болса, онда ол осы класқа тиесілі. Орталық объектілер ретінде алдағы уақытта құрылушы класс центрі болып тағайындалушы кейбір бастапқы аймақты сипаттаушы центрлер – кандай да бір ретпен ерекшеленіп алынған бірлік объектілерін аламыз. Басқа объектілер жиындары бар, олар ақпараттық емес сипаттағы және шеткі жиындар, олар осы кластардың басқа объектілерімен тығыз қоршала орналасқан. Ал оларды тандамалы объектілерден алып тастасақ классификациялау сапасына әсері болмайды. Сонымен қатар таңдамалы объектілер арасында кейбір шулы, шығынды объектілері де болуы мүмкін. Ал оларды алып тастау классификациялау сапасын арттыра түседі.

Осыдан шулы тандамалар мен ақпараттық емес объектілерді алып тастап, тек объектілердің минималды санын ғана қалтыру керектігі туралы тұжырымға келеміз. Нәтижесінде классификациялаудың сапасы, тұрақтылығы артады, деректер көлемі қысқарады, классификациялау уақыты қысқарады.. Орталық объектілерді бастапқы жиынның орталықтанған сипаттамасын ерекшелеп көрсетуге арналған алгоритмдер осы орталық жиындарға сәйкес орталық объектілерді де табады. Мұндай алгоритмдер класын *орталық алгоритмдер класы* деп атайды. Кез-келген орталық алгоритмдерге айтарлықтай қарапайым түрлендірулер жасау арқылы орталық алгоритмге айналдыруға болады.

2-типтегі кластеризациялау алгоритмдері деп $S_1,S_2,\ldots,S_m$ берілген объектілерінің $J_m(S)$ сипаттамаларын $K(\|\alpha_{ij}\|_{m\times l})$, $\alpha_{ij} \in \{0,1\}$, $i=1,2,\ldots,m$, $j=1,2,\ldots,l$ эквивалентті класқа ауыстыру $A(J_m(S))=K(\|\alpha_{ij}\|_{m\times l})$ алгоритмін

қарастырайық. $I = \|\alpha_{ij}\|_{m\times l}$ ақпараттық матрицасы мен $H=(\beta_1,\beta_2,\ldots,\beta_m)$-ы нақты кластеризациялау деп аталады және A арқылы белгінеді.

Ендеше осы алгоритмдер жиындарының, яғни объектілер шоғырларының алгоритмдері жиынында нәтижелік кластарға бөлулерді түзетін құрулардың топтық шешімдер есебін қарастырайық.

Айталық $A^{sh} = \{A_1^{sh}, A_2^{sh}, \ldots, A_n^{sh}\}$-орталық алгоритмдер жиыны болсын. Мұндағы әрбір алгоритм берілген объектілер тобынан ішкі объектілер тобын, яғни орталық объектілер тобының жиынын құрады:

$$A_i^{sh}(S) = S_i^{sh}, \ i = 1,2,\ldots,m$$

мұндағы S_i^{sh}- орталық объектілер жиыны.

$$S_i^{sh} = \{S_{i1}, S_{i_2}, S_{i_3}, \ldots, S_{i_b}\}, \ 1 \le b \le u.$$

Ал егер вектордың бірлік компоненті номері $S=\{S_1,\ldots,S_u\}$ жиынының номерімен сәйкес келсе $H(A_i^{sh})$ векторы A_i^{sh} алгоритмінде $S=\{S_1,\ldots,S_u\}$ объектілер жиыны ортасының сипаттамасы деп аталады да $H'(A_i^{sh})$ арқылы белгілейміз.

$H'(A_i^{sh})$ орталар сипаттамасынан берілген объектілер жиынына A_i^{sh} алгоритмдерін қолдану арқылы $I^{sh} = \|\alpha_{ij}\|_{m\times l}$ ақпараттық матрицасын аламыз және ол орталар сипаттамасының ақпараттық матрицасы деп аталады.

$$A^{sh}(S) = (H_1', H_2', \ldots, H_n'), \quad H_i' = (\beta_1, \beta_2, \ldots, \beta_u), \beta_{ij} = \{0,1\}$$

мұндағы $H_i', \ i = 1,2,\ldots,n$ векторлар жиынынан элементтері 0 мен 1 мәнінен тұратын келесі матрицаны аламыз:

$$\triangle_{ij} = \begin{cases} 1, \text{егер } S_j \epsilon S_i^{sh}, i = 1,2,\ldots,t \\ 0, \quad \text{басқа жағдайларда} \end{cases} \qquad (2.11)$$

Берілген (2.11)-формуладан байқайтынымыз, $\triangle_{ij}=1$ болса, онда A_i^{sh} алгоритмі арқылы S_j объектісі алдағы уақытта құрылушы класс ортасы ретінде алынады.

Енді осы A^{sh} – орталық алгоритмдер жиынында S объектісін S_j^{sh}- j=1,2,...,n ішкі жиындарына S_i объектісінің кіру саны ретінде бағалауын қарастырайық. Объектіні бағалауды A_Γ^{sh}-арқылы белгілейік.

$$A_\Gamma^{sh} = \sum_{j=1}^{n} \triangle_{ij}, i = 1,\ldots,u \qquad (2.12)$$

Осы (2.12)-теңдеуде салмақтық коэффициенттерді ескере отырып келесі теңдеуді аламыз:

$$A_{\Gamma}^{sh} = w \sum_{j=1}^{n} \triangle_{ij}, i = 1, \dots, u$$

Орталар сипаттамаларының жалпыланған $H_b' = (b_1, b_2, \dots, b_u)$ векторын төмендегідей жазамыз:

$$H_b = \begin{cases} 1, & \text{егер } A_{\Gamma}^{sh}(S_i) \geq \theta,\ 1 \leq \theta \leq n, i = 1, \dots, u \\ 0, & \text{басқа жағдайларда} \end{cases}$$

мұндағы, θ-шектеу.

Осы жерден шығатын қорытынды объектілер жиынына орталар алгоритмдерінің тобын қолдана отырып, орталар сипаттамасының жалпыланған векторын аламыз және ол A^{opm} топтық алгоритмдер арқылы бізге қажетті іздеп отырған нәтижелік бөлуімізді береді. Осы берілген әдіс бойынша жасалынған тәжірибеге тоқтала кетейік. Алдымен $S=\{S_1, S_2, \dots, S_u\}$ (u=61) объектілер тобы берілсін, яғни матрица өлшемі 6х61. Осы берілген объектілер тобына толық байланыс (максимин) алгоритмі мен К-ішкі топтық орталар алгоритмдерін қолданып, кластарға бөліп аламыз. Топтық шешімдер алгоритмдерінің саны 6-ға тең: $A_{sh} = \{A_1, A_2, A_3, A_4, A_5, A_6\}$.

1-қадам: базалық алгоритмдерді қолданумен орталық объектілер жиынын құрамыз, A_{sh} алгоритмдері жиынымен бастапқы кластерлер санын өзгерте отырып, нәтижелер аламыз.

2-*қадам:* Алған нәтижелерді матрицаға түрлендіреміз.

3-қадам: Матрицадан класс центрлерін тауып, осы центрге жақын объектіні іздей отырып, ақпараттық матрицаны құрамыз.

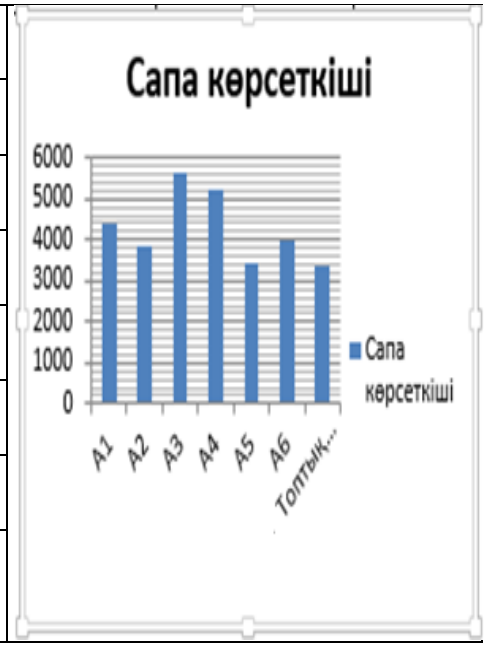
4-қадам: Элементтері *0* мен *1* ден тұратын ақпараттық матрицаны аламыз, мұнда центрге жақын объектілер 1-ге тең, ал қалғандары 0-ге тең.

5-қадам: Матрица бағандары бойынша элементтерін қосындылай отырып, орталар сипаттамаларының жалпыланған векторын аламыз. Нәтижесінде элементтері *0* мен *1* ден тұратын массив құрастырамыз. Біздің жағдайда θ-шектеуі 3-ке тең деп аламыз да, осы құрылған массив бойынша элементтері 3-ке тең немесе үлкен болғанша жүріп өтеміз. Егер элемент 3-тен үлкен немесе тең болған жағдайда бұл элементті ортасы ретінде аламыз. Ары қарай анықталған барлық орталарды k-means алгоритміне береміз. Нәтижесінде объектілерді қажетті бөлуді аламыз.

Объектілерді кластерлерге бөле отырып, қолданылған әрбір алгоритмнің сапа көрсеткішін анықтау жүргізілді. Алынған нәтижелерден байқайтынымыз, жеке алгоритмдерге қарағанда топтық шешімдер алгоритмі жақсы нәтиже береді. Оны төменде келтірілген 2.2-кестесінен байқауға болады. Кестеде қолданылған базалық алгоритмдер, осы алгоритмдер нәтижелеріне қолданылған сапа көрсеткіштерінің нәтижелері, әр алынған нәтижеде пайда болған нәтижелік кластерлер саны, сонымен қатар графиктік түрде көрсетілген сапа көрсеткіштерінің нәтижелері берілген.

Кесте 2.2 – Әрбір алгоритм бойынша алынған нәтижелер

Алгоритмдер	Сапа көрсеткіші	Кластерлер саны
A_1	4396.744572960121	3
A_2	3874.056926864209	4
A_3	5667.491542070022	2
A_4	5199.895200143007	6
A_5	3431.520184595646	7
A_6	4003.474067186382	5
Топтық шешімдер алгоритмі	3381.462215331314	

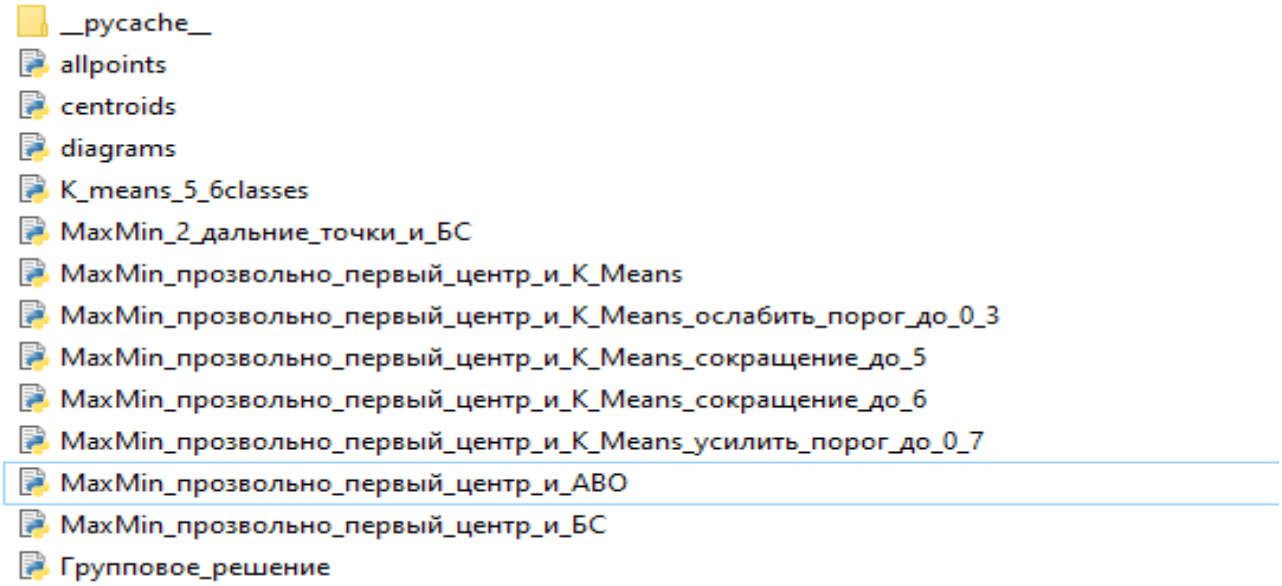


Диссертациялық жұмысты орындау барысында орталық объектілерді оқшаулауға тағы бір тәжірибе жүргізілді. K-means алгоритмінің бастапқы параметрін өзгерте отырып және MaxMin алгоритмінің центрін кездейсоқтық ретпен ала отырып берілген объектілер жиынына кластерлеу жүргізілді. MaxMin алгоритмі арқылы алынған центрлер K-means алгоритміне бастапқы параметрі, яғни центрлер ретінде берілді.

Алгоритмнің белгіленуі	*Алгоритм*	*Қосымша*
A_1	K-means алгоритмі	5-кластық
A_2	K-means алгоритмі	6-кластық
A_3	MaxMin - 2_алыс_нүкте	ЖК
A_4	MaxMin - кездейсоқ_бірінші_центр	K-means
A_5	MaxMin - кездейсоқ_бірінші_центр	K-means_шектеуді 3 ке дейін алу
A_6	MaxMin - кездейсоқ_бірінші_центр	K-means_шектеуді 5 ке дейін алу
A_7	MaxMin - кездейсоқ_бірінші_центр	K-means_шектеуді 6 ке дейін алу
A_8	MaxMin - кездейсоқ_бірінші_центр	K-means_шектеуді 7 ке дейін алу
A_9	MaxMin - кездейсоқ_бірінші_центр	БЕА
A_{10}	MaxMin - кездейсоқ_бірінші_центр	ЖК

Сурет 2.7 – Топтық шешімдерде қолданылған алгоритмдер

Бұл жерде K-means алгоритміне әр түрлі шектеулер 3, 4, 5, 6, 7-ге тең деп қойылды. Сонымен қатар центрлер бағаларды есептеу алгоритміне және жақын көрші әдісіне беріле отырып базалық алгоритмдер тобын құрастырып алдық. Осы модификациялау нәтижесінде алынған алгоритмдердің әрқайысынан жеке-жеке нәтижелер ала отырып, осы нәтижелерді топтық шешімдерді алудың бір тұжырымдамасы орталық объектілерді оқшаулау тәсілін қолдана отырып бір топтық шешімге топтастырдық (2.7-сурет).



Сурет 2.8 – Программа коды сақталынған программалық файлдар

Бұл жерде ол тек деректердің толық мәндерін шығару үшін қолданылды. Сонымен қатар көп өлшемді массивтер, матрицалармен және де осы массивтермен операциялар жасауға арналған математикалық функциялардан тұратын Python программалау тілінің Numpy библиотекасы да қолданылды.

```
import sys
import numpy as np
from allpoints import df
from K_means_5_6classes import unique_centroids_algorithm_1_centrolized, unique_centroids_algorithm_2_centrolized, \
    unique_c_a_mathlib_data_algo1, cluster_counter_algo1, unique_c_a_mathlib_data_algo2, cluster_counter_algo2, \
    cqp_algo1, cqp_oa_algo1, cqp_algo2, cqp_oa_algo2
from MaxMin_прозвольно_первый_центр_и_БС import unique_centroids_algorithm_03, unique_c_a_mathlib_data_algo3, cluster_counter_algo3, cqp_algo3, \
    cqp_oa_algo3
from MaxMin_2_дальние_точки_и_БС import unique_centroids_algorithm_04, unique_c_a_mathlib_data_algo4, cluster_counter_algo4, cqp_algo4, \
    cqp_oa_algo4
from MaxMin_прозвольно_первый_центр_и_K_Means_сокращение_до_6 import unique_centroids_algorithm_05, unique_c_a_mathlib_data_algo5, cluster_counter_algo5, cqp_algo5, \
    cqp_oa_algo5
from MaxMin_прозвольно_первый_центр_и_K_Means_сокращение_до_5 import unique_centroids_algorithm_06, unique_c_a_mathlib_data_algo6, cluster_counter_algo6, cqp_algo6, \
    cqp_oa_algo6
from MaxMin_прозвольно_первый_центр_и_K_Means import unique_centroids_algorithm_07, unique_c_a_mathlib_data_algo7, cluster_counter_algo7, cqp_algo7, \
    cqp_oa_algo7
from MaxMin_прозвольно_первый_центр_и_K_Means_ослабить_порог_до_0_3 import unique_centroids_algorithm_08, unique_c_a_mathlib_data_algo8, cluster_counter_algo8, cqp_alg
    cqp_oa_algo8
from MaxMin_прозвольно_первый_центр_и_K_Means_усилить_порог_до_0_7 import unique_centroids_algorithm_09, unique_c_a_mathlib_data_algo9, cluster_counter_algo9, cqp_algo
    cqp_oa_algo9
from MaxMin_прозвольно_первый_центр_и_АВО import unique_centroids_algorithm_10, unique_c_a_mathlib_data_algo10, cluster_counter_algo10, \
    cqp_algo10, cqp_oa_algo10
```

Сурет 2.9 – сурет. Қолданылған библиотекалар тізімі

Енді table_of_centroid_lab06 функциясы көмегімен алгоритмдер арқылы центрлерді анықтаймын, оны барлық алгоритмдердің элменттері туралы деректер сақталынатын құрылған кестеде 1 ретінде белгілеп отырдым. Ары қарай әр элемент қанша рет центр болғандығын есептейміз (2.9- сурет).

```
lab06_matched_table = np.zeros(shape=lab06_dfs.shape[0], dtype=float)
for matcher_index in range(61):
    for matcher_row in range(10):
        match_counter = 0
        if lab06_table[matcher_row][matcher_index] == 1:
            match_counter += 1
            lab06_matched_table[matcher_index] += match_counter

print("ALL MATCHES")
print(lab06_matched_table)
```

Сурет 2.10 – Центрлерді есептеу

Одан кейін центрлерге арналған шектеу белгілеп аламыз және мұнда берілген шектеу центрлердің қосындысының максималды санынан артпау керек. Енді топтық шешімдердің центрлерін k-means алгоритміне бере отырып осы центрлер арқылы кластерлер құрамыз.

```
while True:
    delta_for_match = int(input("please write the delta > "))
    if delta_for_match > np.max(lab06_matched_table):
        print("Please provide delta less than maximum value of matched table")
    else:
        break
unique_centroids = np.zeros(shape=14, dtype=float)
unique_centroids = [unique_centroids]
for i, each_matched_element in enumerate(lab06_matched_table):
    if each_matched_element == np.max(lab06_matched_table):
        unique_centroids = np.append(unique_centroids, [df[i]], axis=0)
unique_centroids = unique_centroids[~(unique_centroids == 0).all(1)]
count_to_get_through = 0
for each_matched_element in lab06_matched_table:
    if each_matched_element >= delta_for_match:
        count_to_get_through += 1
if count_to_get_through >= 6:
    for i, each_matched_element in enumerate(lab06_matched_table):
        if each_matched_element >= delta_for_match:
            unique_centroids = np.append(unique_centroids, [df[i]], axis=0)
else:
    print("Your delta is too small, programme will populate by slightly decreasing it till it gets at least 6 centroids")
    decrease_still_not_six = 0
    while unique_centroids.shape[0] < 6:
        decrease_still_not_six += 1
        for i, each_matched_element in enumerate(lab06_matched_table):
            if each_matched_element >= delta_for_match:
                unique_centroids = np.append(unique_centroids, [df[i]], axis=0)
                unique_centroids = np.unique(unique_centroids, axis=0)
        for i, each_matched_element in enumerate(lab06_matched_table):
            if (each_matched_element >= (delta_for_match - decrease_still_not_six)) and not(any(np.equal(unique_centroids,df[i]).all(1))) and
            (unique_centroids.shape[0] < 6):
                unique_centroids = np.append(unique_centroids, [df[i]], axis=0)
                unique_centroids = np.unique(unique_centroids, axis=0)

unique_centroids = np.unique(unique_centroids, axis=0)
unique_centroids = unique_centroids[~(unique_centroids == 0).all(1)]
```

Сурет 2.11 – Шектеулер беру

```
def distances_finder(task, unique_centroids, df):
    for indu, xy in enumerate(df):
        inner_dictro_for_d = np.zeros(shape=(unique_centroids.shape[0], 1), dtype=float)
        inner_dictro_for_c = np.zeros(shape=(unique_centroids.shape[0], 14), dtype=float)
        inner_dictro_for_xy = np.zeros(shape=(unique_centroids.shape[0], 14), dtype=float)
        for i, c in enumerate(unique_centroids):
            d = math.sqrt(
                pow((xy[0] - c[0]), 2) + pow((xy[1] - c[1]), 2) + pow((xy[2] - c[2]), 2) + pow((xy[3] - c[3]),
                                                                                               2) + pow(
                    (xy[4] - c[4]), 2) + pow((xy[5] - c[5]), 2) + pow((xy[6] - c[6]), 2) + pow((xy[7] - c[7]),
                                                                                               2) + pow(
                    (xy[8] - c[8]), 2) + pow((xy[9] - c[9]), 2) + pow((xy[10] - c[10]), 2) + pow((xy[11] - c[11]),
                                                                                                 2) + pow(
                    (xy[12] - c[12]), 2) + pow((xy[13] - c[13]), 2))
            inner_dictro_for_d[i] = d
            inner_dictro_for_c[i] = c
            inner_dictro_for_xy[i] = xy
        inner_dict_for_d_xy = np.append(inner_dictro_for_d, inner_dictro_for_xy, axis=1)
        inner_dict_for_d_xy_c = np.append(inner_dict_for_d_xy, inner_dictro_for_c, axis=1)
        if task == 1:
            for index_for_dc_by_x in inner_dict_for_d_xy_c:
                if np.amin(inner_dictro_for_d) == index_for_dc_by_x[0]:
                    overall_dictro[indu] = index_for_dc_by_x
        elif task == 3:
            for index_templ in inner_dict_for_d_xy_c:
                if np.amin(inner_dictro_for_d) == index_templ[0]:
                    overall_dictro3[indu] = index_templ
```

Сурет 2.12 – Кластар құру

Мұнда distance_finder функциясы арқылы центрлерден объектілерге дейінгі ара қашықтықтарды есептеп, табылған деректерді белгілеп жазып отырамыз.

```
def iterator_for_k_means(unique_centroids, df):
    counter_of_iterations = 0
    while True:
        counter_of_iterations += 1
        distances_finder(1, unique_centroids, df)
        for i, q in enumerate(unique_centroids):
            new_centroid_x0 = 0
            new_centroid_sum_x0 = 0
            new_centroid_x1 = 0
            new_centroid_sum_x1 = 0
            new_centroid_x2 = 0
            new_centroid_sum_x2 = 0
            new_centroid_x3 = 0
            new_centroid_sum_x3 = 0
            new_centroid_x4 = 0
            new_centroid_sum_x4 = 0
            new_centroid_x5 = 0
            new_centroid_sum_x5 = 0
            new_centroid_x6 = 0
            new_centroid_sum_x6 = 0
            new_centroid_x7 = 0
            new_centroid_sum_x7 = 0
            new_centroid_x8 = 0
            new_centroid_sum_x8 = 0
            new_centroid_x9 = 0
            new_centroid_sum_x9 = 0
            new_centroid_x10 = 0
            new_centroid_sum_x10 = 0
            new_centroid_x11 = 0
            new_centroid_sum_x11 = 0
            new_centroid_x12 = 0
            new_centroid_sum_x12 = 0
            new_centroid_x13 = 0
            new_centroid_sum_x13 = 0

            count_centroids_points = 0.0
            for j, b in enumerate(overall_dictro):
                if unique_centroids[i][0] == overall_dictro[j][15] and unique_centroids[i][
                    1] == overall_dictro[j][
                    16] and \
```

Сурет 2.13 – k-means алгоритмінің итерациялау

Орталар координаталарын орталарға ең жақын объектілер ретінде жазылып сақталынған центрлермен салыстырамыз. Егер координаталар тең болса, онда координаталарды қосындылап, оларды сақтаймыз, counter count_centroid_points функциясы көмегімен санын есептейміз. Ары қарай count_centroid_points нөлге тең болмаса табылған координаталар қосынидысын объектілер санына бөлеміз. Мұнда уақытша центрлер координаталарын таптық. Енді уақытша центрлер координаталары осы уақытқа дейінгі центрлермен тең бе екендігін тексереміз. Егер олар тең болса уақытша центрлердің итерация саны мен парамерлерін шығарамыз. Егер count_centroid_points нөлге тең болмаса бұрынғы центрлерді жаңа центрлерге теңестіреміз. Бірақ біз уақытша центрлерді нағыз центр ретінде қолдана алмаймыз, сондықтан осы уақытша центрлерге жақын объектілерді табамыз

да оны нағыз центр ретінде аламыз. Осыдан кейін осы центрді пайдалана отырып, барлық объектілерді жақын көрші әдісі бойынша кластарға бөлеміз.

```
def algorithm_12_centroid_perforator(unique_centroids, df):
    overall_dictro2 = np.zeros(shape=(unique_centroids.shape[0], 29), dtype=float)
    distances_finder_resulting(unique_centroids, overall_dictro2, df)
    for i, kendo in enumerate(overall_dictro2):
        bruh = np.zeros(shape=14, dtype=float)
        unique_centroids[i] = 0
        for fourteen in range(14):
            bruh[fourteen] = kendo[fourteen + 1]
        unique_centroids[i] = bruh
    return unique_centroids
```

Сурет 2.14 – кластерлер саны 5 және 6 болатын K-Means алгоритмін есептеу

Бұл жерде (2.14-суретте) algorithm 12 centroid perforator() функциясында топтық шешімдер алгоритмінде қолданылған кластерлер саны 5 және 6 болатын алгоритмдерін есептеу жүргізіліп, келесі суретте нәтижелер көрсетілген.

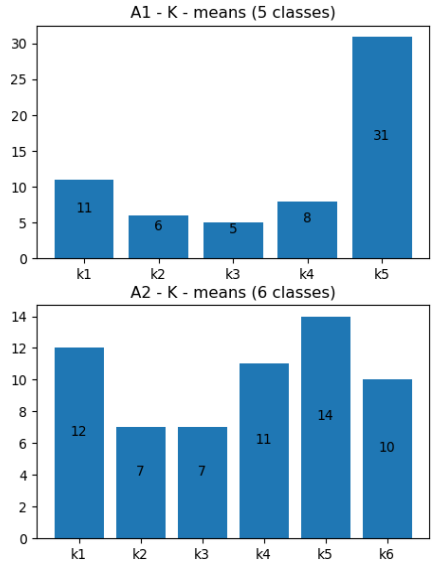
```
CLUSTERS

1) Centroid :
[  0.075   0.59   13.9     0.17   14.3     2.      0.13  459.58   45.96
   2.88  410.74    0.01    0.01    0.89 ]

1) [  0.147   0.75   18.1     0.04   20.3     5.      0.2   520.59   52.06
  39.59  428.94    0.09    0.08    0.82 ]

2) [  0.111   0.6    17.6     0.04   17.5     3.      0.19  418.48   41.85
  -1.08  377.71    0.      0.      0.9  ]

3) [  0.086   0.6    17.6     0.04   20.7     3.      0.14  426.09   42.61
  33.47  350.01    0.1     0.08    0.82 ]

4) [  0.124   0.6    17.6     0.04   19.      3.      0.21  397.4    39.75
  15.12  342.54    0.04    0.04    0.86 ]

5) [  0.106   0.59   13.9     0.17   17.7     2.      0.18  410.68   41.07
  27.35  342.26    0.08    0.07    0.83 ]

6) [  0.075   0.59   13.9     0.17   14.3     2.      0.13  459.58   45.96
   2.88  410.74    0.01    0.01    0.89 ]

2) Centroid :
[  0.173   0.75   18.1     0.04   33.1     5.      0.23  419.42   41.94
 269.94  107.53    2.51    0.64    0.26 ]

1) [  0.208   0.9    15.9     0.04   36.1     5.      0.23  501.09   50.11
 363.52   87.46    4.16    0.73    0.17 ]

2) [  0.173   0.75   18.1     0.04   33.1     5.      0.23  419.42   41.94
 269.94  107.53    2.51    0.64    0.26 ]

3) [  0.191   0.75   18.1     0.04   28.6     5.      0.25  428.95   42.89
 188.96  197.09    0.96    0.44    0.46 ]

4) [  0.197   0.75   18.1     0.04   36.2     5.      0.26  375.74   37.57
 325.73   12.43   26.43    0.87    0.03 ]

5) [  0.174   0.75   18.1     0.04   33.7     4.      0.23  414.66   41.47
```

Сурет 2.15 – Кластерлерге бөлу нәтижесі

Енді құрылған кластерлер сапасын анықтау үшін әр кластерлер элементтерінен барлық элементтерге дейінгі ара қашықтықтардың қосындысын табамыз. Нәтижелер диаграммалар түрінде ұсынылды. Диаграмаларда кластерлер саны, әр кластың сапасы, әр кластағы элементтер саны көрсетілді.

Тәжірибе нәтижелері

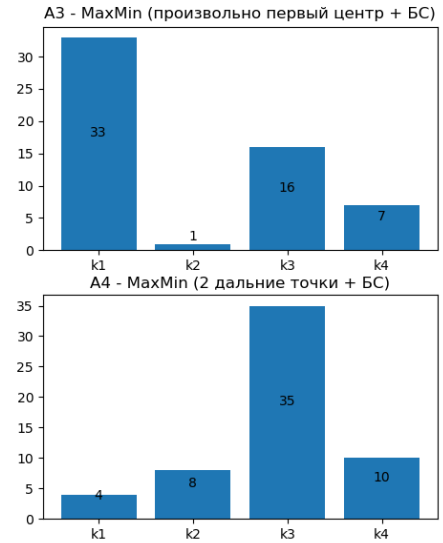
Келесі суретте әр кластер ішіндегі элементтер саны, ал оң жағында центрлер координаталары көрсетілді.



k1--> [0.075, 0.59, 13.9, 0.17, 14.3, 2.0, 0.13, 459.58, 45.96, 2.88, 410.74, 0.01, 0.01, 0.89]
k2--> [0.191, 0.75, 18.1, 0.04, 28.6, 5.0, 0.25, 428.95, 42.89, 188.96, 197.09, 0.96, 0.44, 0.46]
k3--> [0.127, 0.75, 18.1, 0.04, 34.8, 5.0, 0.17, 454.7, 45.47, 300.54, 108.69, 2.77, 0.66, 0.24]
k4--> [0.209, 0.9, 18.1, 0.04, 32.6, 5.0, 0.23, 536.54, 53.65, 260.95, 221.94, 1.8, 0.49, 0.41]
k5--> [0.15, 0.59, 13.9, 0.17, 23.5, 2.0, 0.25, 335.67, 33.59, 69.11, 233.0, 0.3, 0.21, 0.69]

k1--> [0.111, 0.6, 17.6, 0.04, 17.5, 3.0, 0.19, 418.48, 41.85, -1.08, 377.71, 0.0, 0.0, 0.9]
k2--> [0.082, 0.45, 17.3, 0.04, 18.7, 2.5, 0.18, 289.06, 28.91, 12.6, 247.56, 0.05, 0.04, 0.86]
k3--> [0.095, 0.45, 17.3, 0.04, 21.0, 2.5, 0.21, 263.47, 26.35, 33.29, 203.83, 0.16, 0.13, 0.77]
k4--> [0.136, 0.59, 13.9, 0.17, 29.2, 2.0, 0.23, 315.32, 31.53, 110.14, 173.65, 0.63, 0.35, 0.55]
k5--> [0.15, 0.59, 13.9, 0.17, 23.5, 2.0, 0.25, 335.67, 33.59, 69.11, 233.0, 0.3, 0.21, 0.69]
k6--> [0.158, 0.6, 17.9, 0.04, 30.3, 4.0, 0.26, 300.37, 30.04, 178.52, 91.81, 1.94, 0.59, 0.31]

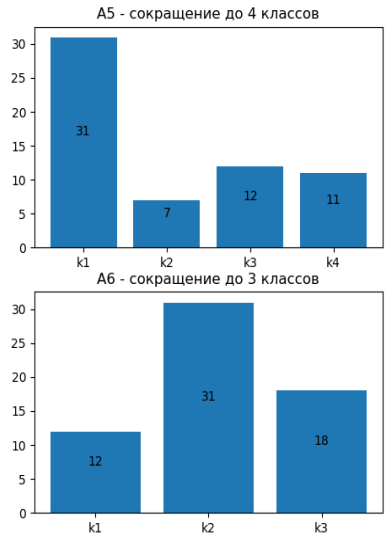
Сурет 2.16 – A_1, A_2 - нәтижелері



k1--> [0.152, 0.59, 13.9, 0.17, 26.6, 2.0, 0.26, 314.77, 31.48, 71.42, 191.87, 0.48, 0.29, 0.61]
k2--> [0.052, 0.6, 17.9, 0.04, 11.0, 4.0, 0.09, 513.05, 51.31, -99.34, 561.09, -0.18, -0.19, 1.09]
k3--> [0.127, 0.75, 18.1, 0.04, 34.8, 5.0, 0.17, 454.7, 45.47, 300.54, 108.69, 2.77, 0.66, 0.24]
k4--> [0.05, 0.5, 13.9, 0.17, 12.0, 2.0, 0.08, 498.34, 43.8, -15.1, 463.6, -0.03, -0.03, 0.93]

k1--> [0.052, 0.6, 17.9, 0.04, 11.0, 4.0, 0.09, 513.05, 51.31, -99.34, 561.09, -0.18, -0.19, 1.09]
k2--> [0.216, 0.75, 18.1, 0.04, 33.8, 5.0, 0.29, 372.03, 37.2, 282.54, 52.28, 5.4, 0.76, 0.14]
k3--> [0.033, 0.45, 17.3, 0.04, 18.5, 2.5, 0.07, 339.14, 33.91, 10.8, 294.43, 0.04, 0.03, 0.87]
k4--> [0.218, 0.9, 15.9, 0.04, 28.8, 5.0, 0.24, 536.46, 53.65, 232.15, 250.67, 0.93, 0.43, 0.47]

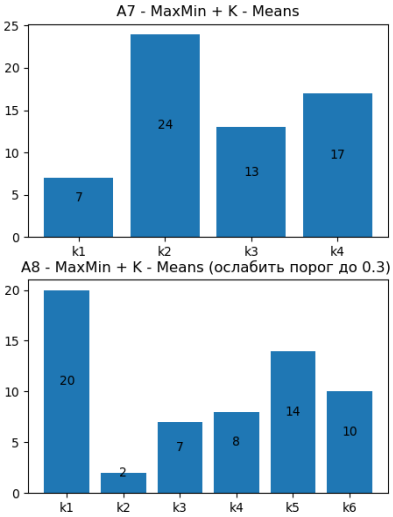
Сурет 2.17 – A_3, A_4 - нәтижелері



k1--> [0.152, 0.59, 13.9, 0.17, 26.6, 2.0, 0.26, 314.77, 31.48, 71.42, 191.87, 0.48, 0.29, 0.61]
k2--> [0.127, 0.75, 18.1, 0.04, 34.8, 5.0, 0.17, 454.7, 45.47, 300.54, 108.69, 2.77, 0.66, 0.24]
k3--> [0.075, 0.59, 13.9, 0.17, 14.3, 2.0, 0.13, 459.58, 45.96, 2.88, 410.74, 0.01, 0.01, 0.89]
k4--> [0.161, 0.75, 17.9, 0.04, 30.6, 4.0, 0.21, 445.54, 45.55, 182.84, 218.15, 0.84, 0.41, 0.49]

k1--> [0.075, 0.59, 13.9, 0.17, 14.3, 2.0, 0.13, 459.58, 45.96, 2.88, 410.74, 0.01, 0.01, 0.89]
k2--> [0.082, 0.45, 17.3, 0.04, 18.7, 2.5, 0.18, 289.06, 28.91, 12.6, 247.56, 0.05, 0.04, 0.86]
k3--> [0.223, 0.9, 15.9, 0.04, 29.2, 5.0, 0.25, 529.06, 52.91, 239.35, 236.81, 1.01, 0.45, 0.45]

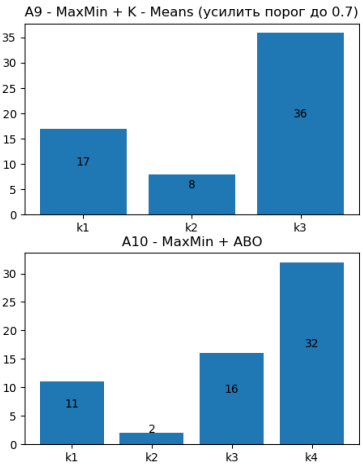
Сурет 2.18 – A_5, A_6 - нәтижелері



k1--> [0.05, 0.5, 13.9, 0.17, 12.0, 2.0, 0.08, 498.34, 43.8, -15.1, 463.6, -0.03, -0.03, 0.93]
k2--> [0.127, 0.59, 13.9, 0.17, 24.1, 2.0, 0.22, 354.25, 35.42, 73.42, 245.4, 0.3, 0.21, 0.69]
k3--> [0.085, 0.45, 17.3, 0.04, 20.1, 2.5, 0.19, 278.43, 27.84, 25.19, 225.39, 0.11, 0.09, 0.81]
k4--> [0.2, 0.9, 19.9, 0.04, 31.8, 5.0, 0.22, 536.21, 53.62, 286.14, 196.45, 1.46, 0.53, 0.37]

k1--> [0.045, 0.45, 17.3, 0.04, 20.0, 2.5, 0.1, 318.98, 31.9, 24.29, 262.79, 0.09, 0.08, 0.82]
k2--> [0.132, 0.75, 18.1, 0.04, 11.5, 5.0, 0.18, 581.87, 58.19, -118.78, 642.46, -0.18, -0.2, 1.1]
k3--> [0.127, 0.75, 18.1, 0.04, 34.8, 5.0, 0.17, 454.7, 45.47, 300.54, 108.69, 2.77, 0.66, 0.24]
k4--> [0.075, 0.59, 13.9, 0.17, 14.3, 2.0, 0.13, 459.58, 45.96, 2.88, 410.74, 0.01, 0.01, 0.89]
k5--> [0.148, 0.6, 17.6, 0.04, 28.5, 3.0, 0.25, 319.25, 31.93, 117.7, 169.63, 0.69, 0.37, 0.53]
k6--> [0.223, 0.9, 15.9, 0.04, 29.2, 5.0, 0.25, 529.06, 52.91, 239.35, 236.81, 1.01, 0.45, 0.45]

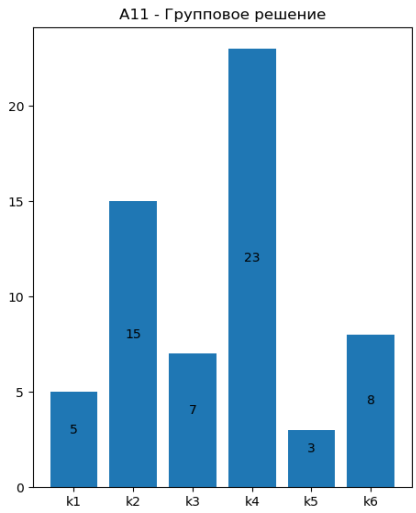
Сурет 2.19 – A_7, A_8 - нәтижелері



k1--> [0.2, 0.9, 19.9, 0.04, 31.8, 5.0, 0.22, 536.21, 53.62, 286.14, 196.45, 1.46, 0.53, 0.37]
k2--> [0.05, 0.5, 13.9, 0.17, 12.0, 2.0, 0.08, 498.34, 43.8, -15.1, 463.6, -0.03, -0.03, 0.93]
k3--> [0.15, 0.59, 13.9, 0.17, 23.5, 2.0, 0.25, 335.67, 33.59, 69.11, 233.0, 0.3, 0.21, 0.69]

k1--> [0.247, 0.6, 17.9, 0.04, 26.6, 4.0, 0.41, 233.47, 23.35, 125.25, 84.87, 1.48, 0.54, 0.36]
k2--> [0.132, 0.75, 18.1, 0.04, 11.5, 5.0, 0.18, 581.87, 58.19, -118.78, 642.46, -0.18, -0.2, 1.1]
k3--> [0.188, 0.9, 15.9, 0.04, 30.8, 5.0, 0.21, 554.36, 55.44, 268.14, 230.78, 1.16, 0.48, 0.42]
k4--> [0.111, 0.6, 17.6, 0.04, 17.5, 3.0, 0.19, 418.48, 41.85, -1.08, 377.71, 0.0, 0.0, 0.9]

Сурет 2.20 – A_9, A_{10} нәтижелері



k1--> [0.138, 0.75, 17.9, 0.04, 17.7, 4.0, 0.18, 542.38, 54.24, -2.88, 491.02, -0.01, -0.01, 0.91]
k2--> [0.106, 0.59, 13.9, 0.17, 17.7, 2.0, 0.18, 410.68, 41.07, 27.35, 342.26, 0.08, 0.07, 0.83]
k3--> [0.173, 0.75, 18.1, 0.04, 33.1, 5.0, 0.23, 419.42, 41.94, 269.94, 107.53, 2.51, 0.64, 0.26]
k4--> [0.152, 0.59, 13.9, 0.17, 26.6, 2.0, 0.26, 314.77, 31.48, 71.42, 191.87, 0.48, 0.29, 0.61]
k5--> [0.187, 0.9, 15.9, 0.04, 33.8, 5.0, 0.21, 536.74, 53.67, 322.13, 160.93, 2.0, 0.6, 0.3]
k6--> [0.218, 0.9, 15.9, 0.04, 28.8, 5.0, 0.24, 536.46, 53.65, 232.15, 250.67, 0.93, 0.43, 0.47]

Сурет 2.21 – топтық шешімдер алгоритмінің нәтижесі

Алгоритмдердің сапа көрсеткіштерін анықтау

Class Qualities

A1 - K - means (5 classes)
L = 5
P(k1) = [8545.] m1 = 11
P(k2) = [1652.22] m2 = 6
P(k3) = [1053.55] m3 = 5
P(k4) = [1465.94] m4 = 8
P(k5) = [49762.06] m5 = 31
F = 62478.76504810176

A2 - K - means (6 classes)
L = 6
P(k1) = [10371.76] m1 = 12
P(k2) = [720.85] m2 = 7
P(k3) = [1330.55] m3 = 7
P(k4) = [7803.26] m4 = 11
P(k5) = [12519.43] m5 = 14
P(k6) = [7696.13] m6 = 10
F = 40441.98214669282

A3 - MaxMin (произвольно первый центр + БС)
L = 4
P(k1) = [62865.14] m1 = 33
P(k2) = [0.] m2 = 1
P(k3) = [16435.45] m3 = 16
P(k4) = [2101.25] m4 = 7
F = 81401.84294670814

A4 - MaxMin (2 дальние точки + БС)
L = 4
P(k1) = [647.93] m1 = 4
P(k2) = [4333.63] m2 = 8
P(k3) = [72987.24] m3 = 35
P(k4) = [4009.65] m4 = 10
F = 81978.44793930628

A5 - сокращение до 4 классов
L = 4
P(k1) = [51103.38] m1 = 31

A7 - MaxMin + K - Means
L = 4
P(k1) = [3208.49] m1 = 7
P(k2) = [30159.88] m2 = 24
P(k3) = [7522.54] m3 = 13
P(k4) = [19231.33] m4 = 17
F = 60122.24127770072

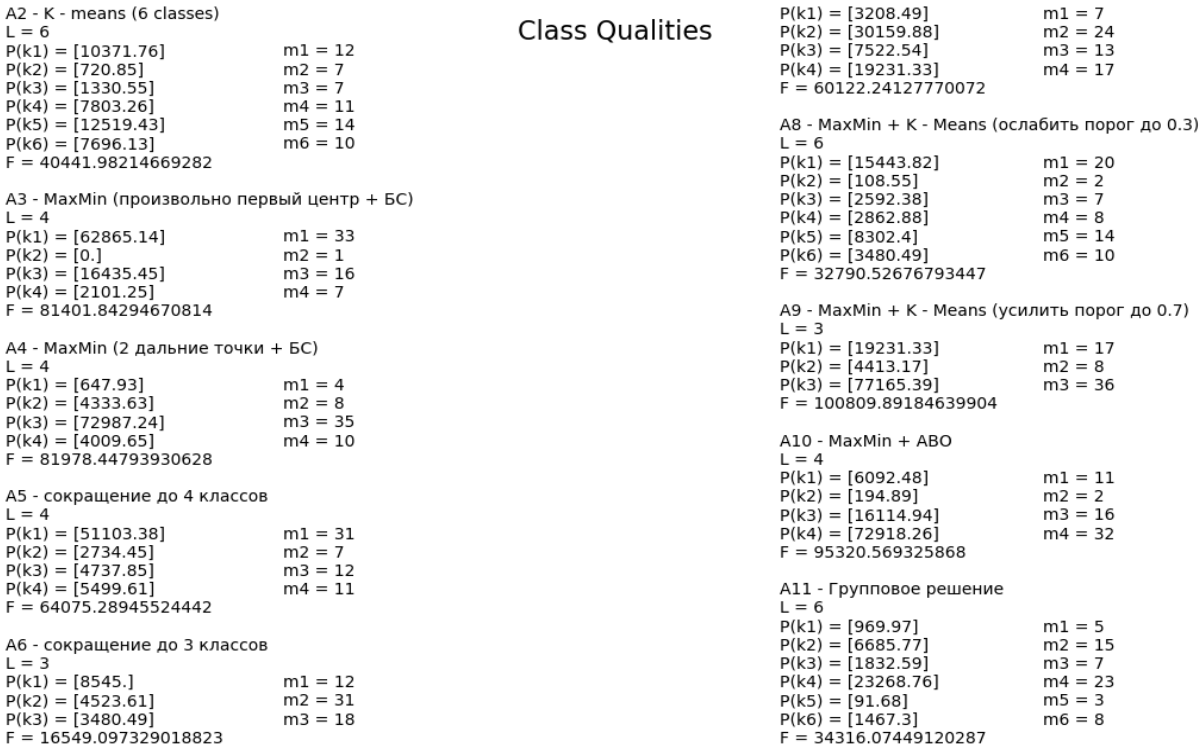
A8 - MaxMin + K - Means (ослабить порог до 0.3)
L = 6
P(k1) = [15443.82] m1 = 20
P(k2) = [108.55] m2 = 2
P(k3) = [2592.38] m3 = 7
P(k4) = [2862.88] m4 = 8
P(k5) = [8302.4] m5 = 14
P(k6) = [3480.49] m6 = 10
F = 32790.52676793447

A9 - MaxMin + K - Means (усилить порог до 0.7)
L = 3
P(k1) = [19231.33] m1 = 17
P(k2) = [4413.17] m2 = 8
P(k3) = [77165.39] m3 = 36
F = 100809.89184639904

A10 - MaxMin + ABO
L = 4
P(k1) = [6092.48] m1 = 11
P(k2) = [194.89] m2 = 2
P(k3) = [16114.94] m3 = 16
P(k4) = [72918.26] m4 = 32

Сурет 2.22 – Алгоритмдер нәтижелері

Берілген суретте барлық алгоритмдердің нәтижелері көрсетілген. Бұл жерде алгоритм атауы жазылып, сонан соң (L) кластерлер саны көрсетілген. Ары қарай алгоритм ішінде әрбір (P[K1]) кластерінің сапасын көруге болады. Мұнда F алгоритмнің жалпы сапасын көрсетеді, функционал мәні минимумға ұмтылған сайын нәтиже жоғары болып табылады. Алгоритмдер нәтижелерінен *A11 топтық шешімдер алгоритмінің* нәтижесі жақсы екендігін көруге болады, яғни бұл берілген әдістің тиімділігін көрсетеді.



Сурет 2.23 – Алгоритмдер нәтижелері

2.6 Кластерлер ядроларын бөліп алумен жинақы жиындарды құру

Объектілерді кластерлерге тиімді нәтижелік бөлуде кластеризациялаудың бірнеше нәтижелерінен топтық шешім алу кластерлік шешім сапасын арттырудағы маңызды процесс. Бірнеше классификациялау алгоритмдерін қолдана отырып жинақы ішкі жиындарды бөліп алуға ядролық әдіс қолданылады. Көптеген тану есептерін шешуші жұмыстарда ядролық әдістерді қолдану кеңінен таралған [57-58].

Қандай да бір бастапқы объектілер тобынан алынған $S=\{S_1, S_2,..., S_u\}$ объектілер тобы берілсін. Берілген объектілер тобы бірнеше кластардан тұрсын және олар бір-біріне эквивалентті деп есептейік. S объектілерін эквивалентті объектілер кластарына берілген $A_1,A_2,...,A_n$ алгоритмдер жиыны арқылы бөлу. Кез-келген классификациялау жүргізуде ақпараттық матрица түріндегі шешім алып, барлық объектілер жұптарын екі ішкі жиындарға ажыратамыз:

$$A_r(J(S))= \left\| a_{ij}^r \right\|_{uxl},\ r=1,2,...,n$$

мұндағы $\left\| a_{ij}^r \right\|_{uxl}$ ақпараттық матрица.

A_r – алгоритмдерімен алынған деректерді қолданып ары қарай жұмыс жасау үшін келесі әдісті қарастырамыз:

$$\{(S_i, S_j)\}^{1r} = (S_i, S_j) | \exists\ K \in \{1,2,...,l_r\},$$
$$i=1,..., m-1, j=i+1,..., m,\ a_{ik}=a_{jk}=1; \qquad (2.13)$$

сонымен қатар

$$\{(S_i, S_j)\}^{2r} = (S_i, S_j) \mid \exists K \in \{1,2,\ldots,l_r\},$$
$$i=1,\ldots, m-1, j=i+1,\ldots, m, \; a_{ik} \neq a_{jk}; \qquad (2.14)$$

(2.13) пен (2.14) берілгендері арқылы $K_1, K_2, \ldots, K_l$ -дың қайсысына S объектілер жұбының тиістілігін анықтайды.

S, $\{(S_i, S_j)\}^{1r}$, $\{(S_i, S_j)\}^{2r}$ жиындары арқылы келесі графтарды тауып алуға болады:

$$\mathcal{F}^{1r} = (S, \{(S_i, S_j)\}^{1r}, \qquad \mathcal{F}^{2r} = (S, \{(S_i, S_j)\}^{2r}$$

мұндағы, $\mathcal{F}^{1r}$ толық ішкі графтардың жиынтығы.

Мұнда кластерлік бөлулер барлық бөлулермен келісілген болуы қажет. Енді осы графтар жиынтығын құру және Q(t) - талдауын құру мәселесін қарастырамыз, мұндағы графтың әр қабырғасы $\mathcal{F}^1, \ldots, \mathcal{F}^t$, $[t/2] \leq Q(t) \leq t$ графының Q(t)-талдауының қасиетінен кем болмайтындай алынған.

Айталық $\mathcal{W}$ – n-өлшемді $w=(w_1,\ldots,w_t)$ бульдік векторлар жиынтығы болсын және де вектор нормасы $\|w\| = \mathcal{R}(t)$ және $\boldsymbol{\tau}(w)=\{i/w_i=1\}$, онда

$$\mathcal{F} = (S, \{(S_i, S)\}) = \cup_{w \in \mathcal{W}} \cap_{r \in \tau(w)} \mathcal{F}^{1r},$$

$$\{(S_i, S_j)\} = \cup_{w \in \mathcal{W}} \cap_{r \in \tau(w)} \{(S_i, S_j)\}^{1r}$$

(S_x, S_r) объектілер жұбына қандай да бір $\gamma_1,\ldots,\gamma_n$ функция енгізсек:

$$\gamma(S_x, S_r) = \begin{cases} 1, \text{егер} (S_x, S_r) \in \{((S, S))\}^{1r}, \\ 0, \text{басқа жағдайларда} \end{cases}$$

олай болса келесі теңдеуді аламыз:

$$\{(S_i, S)\} = \{(S_x, S_r) \mid \sum \gamma_r (S_x, S_r) \geq \mathcal{R}(t)\}.$$

Әр $B_r \subset \{S\}x\{S\}$ объектілердің қатынастары мен $\mathcal{F}^r$ графымен жақындық матрицасы сәйкес келеді: $\|\gamma_r(S_i, S_j)\|_{uxu}$, $r=1,\ldots,n$. Енді оны u^2 өлшемдер векторларындай қарастырайық:

$$\rho(B_x, B_r) = \sum_{i,j=1}^{u} |\gamma_r S_x, S_r) - \gamma_r (S_i, S_j)|$$

бұл жерден енді барлық $B_1 \subset \{S\}x\{S\}$ мүмкін болатын B-ның ең кіші мәнін анықтаймыз:

$$\sum_{r=1}^{n} \rho(B_1, B_r)$$

Онда бастапқы $B_1,\ldots,B_n$ үшін келісілген бинарлық қатынасы $\widetilde{B}$ болып табылады.

Теорема 1. $\mathcal{F}$-графы сәйкесінше келесі төмендегі теңдеудің шартын қанағаттандырса, онда $1\leq Q(t)\leq n$ B бинарлы қатынасын көрсетеді:

$$\sum \rho(B,B_r)=\min_{B^1\subset\{S\}x\{S\}}\sum_{r=1}^{n}\rho(B^1,B_r)$$

Бұл жерде теоремадан түйіндейтініміз бастапқы объектілер тобына топтық алгоритмдерді қолданудағы нәтижелерінен алынған бинарлы қатынастардың орташа мәнімен $\mathcal{F}(\mathcal{R}'(t))$ графының бинарлы қатынасы сәйкес келетін $\mathcal{R}'(t)$ анықталады. Сонымен тиімді $\mathcal{F}(Q(t))$ графын алуда $\mathcal{R}'(t)\geq \rfloor n/2\lfloor+1$ шартын пайдалану жеткілікті болады.

Графтың толықтай бөліктерін табу үшін егер $S_n\subset J(\mathcal{F}_i)$ *және* $S_m\subset J(\mathcal{F}_j)$ болса, онда $J(\mathcal{F}_i)\cap J(\mathcal{F}_j)=0$, $i\neq j, i,j=1,\ldots,l$ шартын қанағаттандыратындай барлық $\mathcal{F}_i$, $i=1,2,\ldots,l$ туынды ішкі графтарды табу қажет, мұнда $(S_n, S_m)\notin \cup(\mathcal{F})$, $J(\mathcal{F}_S)$ – граф төбелері, ал $\cup(\mathcal{F}_S)$ – кездейсоқ графтың қабырғалары.

Анықтама 8. Егер кез-келген S_m төбелеріне келесі $(S_m,S_x)\subset B_C$ және $(S_m,S_r)\subset B_C$ шарттарды қанағаттандыратын кем дегенде екі төбелері табылса $BC\{S\}x\{S\}$ бинарлық қатынас нашар эквивалеттілікте деп атауымызға болады. Бұл жерден $(S_x,S_r)\subset B_S$.

Үш төбеден тұратын толықтай графтың боялған бөліктерін ерекшеленіп алуға байланысты есептерді шешу – қабырғалар бойынша байланыстарға қатысты толықтай эквивалеттілікке нашар эквивалеттілік жуықтау болатын жиындардың нашар эквивалеттіліктегі төбелерінің барлық ішкі жиындарын ерекшелеп алу болып табылады.

Графта құрылымдық өлшемі үшін төбелер мен қабырғаларды ала отырып $\mathcal{F}$ графында монотонды жүйесін анықтау арқылы $\mathcal{F}$ графының нәтижелік $Q(t)$ талдауына қолданылатын ядроны ерекшелеп алу есебіне ұқсас есепті қарастырайық. Ол есепті шешу үшін келесі бір жағдайды қарастырып көрейік: бір-бірімен өзара байланысқан элементтер жиыншаларының орнына $\mathcal{F}$ графының $J(\mathcal{F})$ төбелер жиыны алынса, яғни $S=\{S_1, S_2,\ldots, S_e\}$. Бізде Q - толық бағдарланбаған граф және үш төбесі болсын. Барлық $\tilde{S}\subset S$ үшін бүтін мәнді g_S салмақтық функциясын табайық. $\mathcal{F}$ арқылы $\tilde{S}$ төбелері жиынтығы бар G графының туынды ішкі графын белгілеп алайық. Әртүрлі G графының туынды ішкі графтары санын есептеп, Q изоморфты графты, төбелер санын есептейік. Есептелінген сандар $\tilde{S}$-та анықталған функция мәні ретінде алайық, яғни $f_{\tilde{S}}(S_i)$. Бұл жерде мәндерді есептеу айтарлықтай күрделілікте болмайтындығын көруге болады.

Сонымен $\mathcal{F}$ графында ядроны ерекшелеп алу алгоритмін жазайық.

1 – қадам. Объектілердің бастапқы S объектілер жиынында V^1 төбесі жатады және ол келесі шартты қанағаттандырады:

$$g_S(V_1)=\min_{s\in S} g_s(S)$$

S_{i_1} төбесі {S} объектілер тізбегін анықтаушы алғашқы элемент ретінде алынады. $\tilde{S}^1=S$, $\varepsilon^1=g_3(V_1)$. Енді $g_{s/S} \le \varepsilon^1$ шарты орындалатындай $s \in S$ төбелерін ерекшелеп аламыз да, оларды қандай да бір кез-келген ретпен алынған $\{S\}$ тізбекке қосамыз. Одан кейін $g_{s/\{S\}} \le \varepsilon^1$ шартын қанағаттандыратын төбелердің барлығы $\{S\}$ қосылып отырады, тағы осылай ары қарай шартты тексеру жүргізіледі де кейбір $\{S\} = \{S_1(1),\dots,S_1(i_1)\}$ және S төбесі болмаған жағдайда аяқталады.

Айталық,

$$\varepsilon^s = g_{s/\{S\}}(V^k)) = \min_{s\epsilon S/\{S\}} g_{s/\{S\}}(S) = F\left(\frac{s}{\{S\}}\right),$$
$$S^k = s/\{S\}$$

деп алайық та V^K – ны $\{S\}$ – тізбегімен біріктіріп, оған $g_{s/\{S\}}(S) \le \varepsilon^1$ шарты орындалатындай S төбелерін қосамыз. Ары қарай $g_{s/\{S\}} \le \varepsilon^s$ орындалатындай барлық төбелер таусылғанша тағы шартты тексеріп отырамыз. Қандай да n>s болған жағдайында тізбек барлық жиынды тауысады. Түзілген құрылымдық $\tilde{S}^1, S^2, \dots, \tilde{S}^n$ ішкі жиындардың тізбегіндегі соңғы жиын $\tilde{S}^n$ – қажетті іздеп отырған ядромыз, оның үстіне $S^n \subset \tilde{S}^{n-1} \subset \cdots \subset \tilde{S}^1$, $F(\tilde{S}^n)= \varepsilon^n$.

2.7 Жұптық айырмашылықтар матрицасы

Соңғы уақыттарда кластерлік талдауда шешімдерді топтық қабылдауға негізделген тәсілдер қарқынды түрде даму үстінде. Кластерлік талдаудың әрбір алгоритмі кейбір кіріс параметрлерінен тұрады. Мысалы, кластерлер саны, шекаралық арақашықтық және т.б. Кейбір жағдайларда алгоритм жұмысының қандай параметрлері ең жақсы екендігін белгісіз болады. Осы кезде тек бір нақты параметр ғана емес, бірнеше әртүрлі параметрлері бар алгоритмдерді қолдану керек. Топтық (ансамблдік) тәсіл кластерлеу сапасын жақсартуға мүмкіндік береді. Кластерлік талдауды топтық шешуді құру әдістерінің негізгі бірнеше бағыттары бар: келісе отырып үлестіруге, коассоциативті матрицаға, үлестірулер қоспасы моделіне, теоретикалық-графтық және басқа да әдістерге негізделінген бағыттар [59]. Бұл жұмыста коассоциативті матрицаға негізделген топтық кластерлік талдау әдісі қолданылатын болады. Коасоциативті матрица объектілер жұбының әртүрлі бөлу нұсқаларындағы әртүрлі кластерлерде қаншалықты жиі болатындығын анықтайды.

Орташаланған коассоциативті матрица:

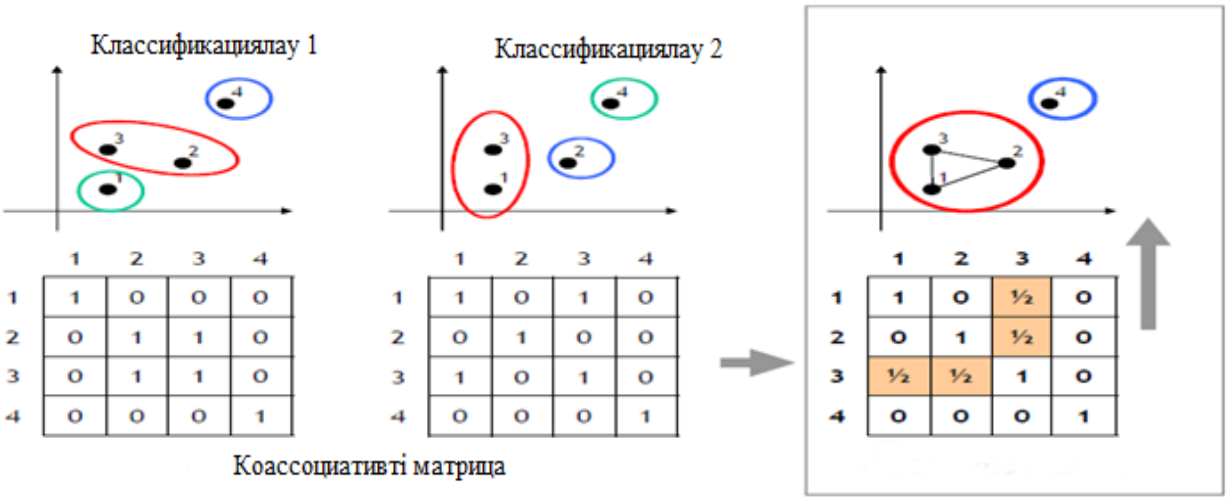
$$M = \sum_{i=1}^{L} w_l M_l \qquad (2.15)$$

мұндағы L – әртүрлі параметрлі алгортмдерді орындау саны;

$$M_l = (m_l(i,j))$$

Егер i және j объектілері 1 – алгоритмде әртүрлі кластерлерге жататын болса $m_l(i,j) = 1$, басқа жағдайда $m(i,j) = 0$, $w_l \geq 0$ алгоримдердің салмағы $\sum_{l=1}^{L} w_l = 1$. Қорытынды келісілген кластерлік бөлуді алу үшін иерархиялық топтауды құрудың агломеративті алгоритмін (дендограммалар құру) қолдануға болады. Топтық шешімде алгоритмнің салмағын анықтау әр-түрлі әдіспен жүреді.

Берілген жұмыста классификациялаудың топтық шешімдері мен ядролық әдістерін біріктіре отырып жаңа классификациялаудың топтық шешімдер алгоритмін алу қарастырылған. Топтық шешімдер алгоритмі басқа алгоритмдерден қарағанда жақсы нәтиже беретіндігі көрсетілген. Алынған алгоритм негізгі екі қадамнан тұрады. Бірінші қадамда кластерлік ансамбльді қолдана отырып, орташаланған коассоциация матрицасын есептеу жүргізіледі, ал екінші қадамда алынған матрицаны кіріс деректері ретінде пайдалана отырып, тиімді классификатор табу жүргізіледі. Коассоциалық матрица рангі толық емес, мұндай матрицамен жасалынатын алгебралық операциялар жұмысы күрделі болмайды. Матрицаны құру үшін $S = \{S_1, S_2, S_3, \ldots, S_m\}$ барлық объектілер жиынын кластерлік талдаудың $A_1, \ldots, A_m$ әртүрлі алгоритмдерімен кластерлеуді жүргізу керек. Әр алгоритм L_m бөлулерді береді, кластерлеуге қолданылған алгоритмдердің нәтижесінде берілген деректер арқылы қажетті матрицаны құрып аламыз. Коассоциациалық матрицаны құру жолы берілген (2.24-сурет) суретте көрсетілген.



Сурет 2.24 – Коассоциалық матрицаның құрылу жолы

Диссертацияда гиперспектралді кескіндерге кластерлік талдау жасау үшін ұсынылған тәсіл бойынша (2.15)-формуласымен коассоциалық матрицасын құрылды. Объектілер жұбы қаншалықты жиі бір класқа тиісті болса, соншалықты біз оларды ұқсас деп санаймыз [60].

Алынған H орталанған жұптық айырмашылықтар матрицасы классификациялаудың ядролық әдістерінде ядро ретінде қолданылады. Сонымен SVM кластерлік талдау алгоритмі мен классификациялаудың ядролық әдістерін үйлестіру арқылы келесі топтық шешімдер алгоритмін алдық.

Классификациялау есептерін шешуде ядролық әдістерді қолдану кең таралуда. Бұл әдіске жататын классификациялаудың SVM танымал әдісін қарастырайық. SVM әдісі бинарлы классификациялау болып табылады. Бұл әдісте екі класқа бөлуде (бинарлы классификациялау есебі) $K = \{k_1, \ldots, k_N\}$, класты $S = \{s_1, \ldots, s_N\}$ объектілерінің деректері беріледі, мұндағы $k_i \epsilon \{+1, -1\}$, $i=1, \ldots, n$ нүкте ретінде болады және оларды $(m-1)$ өлшемді гипержазықтығымен, бөлуші жазытықтың екі класқа дейінгі арақашықтығы максималды болатындай ала отырып бөлу керек. Тірек векторлар әдісінің міндеті тиімді бөлуші гипержазықтықты құру болып табылады. Бөлуші сызық шетінде жатқан нүктелер тірек векторлары деп аталады. Гирпежазықтық теңдеуі $(w, s) + b = 0$ екені белгілі, мұндағы w – бөлуші гипержазықтыққа перпендикуляр вектор, ал b көмекші параметр. Тірек векторының әдісі келесі түрдегі шешуші функцияны құрады:

$$F(s) = sing\left(\sum_{i=1}^{n} \lambda_i c_i(s_i, s) + b\right) \quad (2.16)$$

Бұл жерде S объектілері $F(s)=1$ мен бір класқа жатқызылады, ал $F(s)=0$ мен басқа класқа жатқызылады. $F(s)$ шешуші функциясы объектілердің скаляр көбейтіндісіне тәуелді. Сондықтан (s, s) скаляр көбейтіндісін $(\varphi(s_i, s))$ түріндегі көбейтіндімен алмастыруға болады:

$$\mathrm{F(s)} = \mathrm{sing}\left(\sum_{i=1}^{n} \lambda_i c_i(\varphi(s_i), \varphi(s)) + b\right) \quad (2.17)$$

Мұнда $\mathrm{K(s, s')} = (\varphi(s_i), \varphi(\mathrm{s}'))$ функциясы ядро деп аталады. Ядроны таңдау түзетуші кеңістікке өтуге анықтайды және сызықты бөлінбейтін берілгендерге классификациялаудың сызықты алгоритмдерін қолдануға мүмкіндік береді [61].

Топтық шешімдер арқылы алынған алгоритм

Берілген диссертациялық жұмыста ұсынылып отырған алгоритмнің негізгі идеясы (2.15) – коассоциативті матрицасын құрудан тұрады. Бұл матрица бастапқы берілген объектілер жиынына кластеризациялаудың әртүрлі алгоритмдерін қолдана отырып алынады. Мұнда объектілер бірдей кластерлерге түскен сайын оларды бір-біріне ұқсас деп аламыз. Алгоритм келесі қадамдардан тұрады:

Берілгендер: K_c класы берілген S_c объектілері мен S_u объектілері, M кластеризациялау алгоритмдерінің саны, әрбір μ_m, $m=1, \ldots, M$ алгоритмдерімен L_m кластеризациялау саны.

Нәтиже: S_u объектілерінің кластары.

1. S_c және S_u объектілеріне $\mu_1,\ldots,\ \mu_m$ кластерлік талдау алгоритмдері арқылы кластерлеу жүргіземіз, әр $\mu_m, m = 1,\ldots,M$ алгоритмдерінен объектілерге бөлудің L_m нұсқасын аламыз.
2. (2.15)-формула бойынша H матрицасын есептейміз.
3. H матрицасын ядро ретінде алып SVM-ді S_c деректерімен оқытамыз.
4. SVM көмегімен S_u объектілері үшін кластарды көрсету керек.

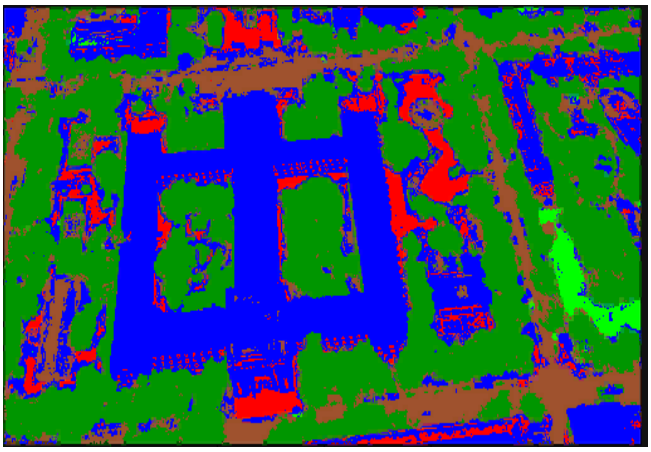
Алгоритмнің соңы.

Алынған алгоритмді тәжірибелік зерттеу үшін спутниктен түсірілген кескіндер қолданылды.

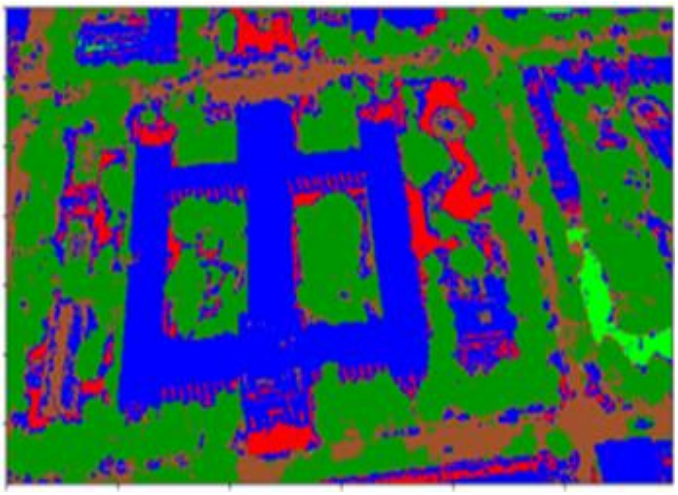
2.8 Эксперименттік тәжірибе нәтижесі

Әдетте RGB-бейнесі үш каналдан тұрады: әр түстер бойынша қанықтық мәні. Кей жағдайларда бұл түсірілген объект сипаттамасы туралы толық ақпарат алуға жеткіліксіз болады. Адам көзімен айыру мүмкін емес объектілердің қасиеттері туралы деректерді алу үшін гиперспектральды бейне қолданылады. Алынған алгоритмді тәжірибелік зерттеу үшін өлшемі 610 да 340 пикселді, 103 спектрлі каналдан тұратын Ғылым ордасы және Сүлеймен Демирел атындағы Университеті кескіні қолданылды.

Сурет 2.25 – Ғылым ордасы кескінінің гиперспектральды бейнесі

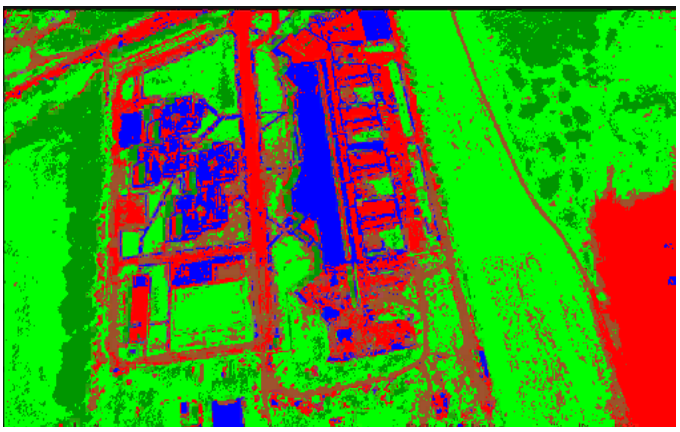


Сурет 2.26 – Топтық шешімдер алгоритмінің нәтижесі



Сурет 2.27 – XGBoost алгоритмінің нәтижесі

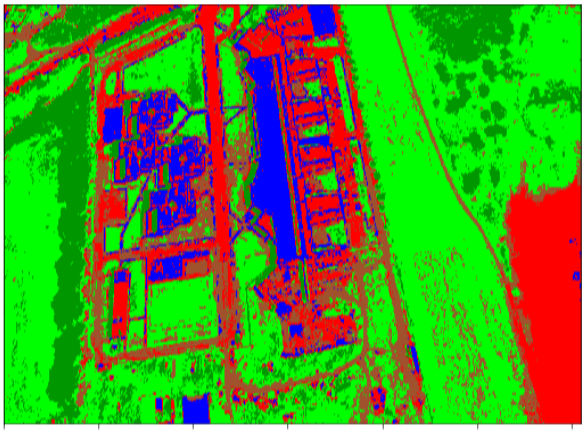
Сурет 2.28 – Сүлеймен Демирел атындағы Университеті кескінінің гиперспектральды бейнесі



Сурет 2.29 – Топтық шешімдер алгоритмінің нәтижесі

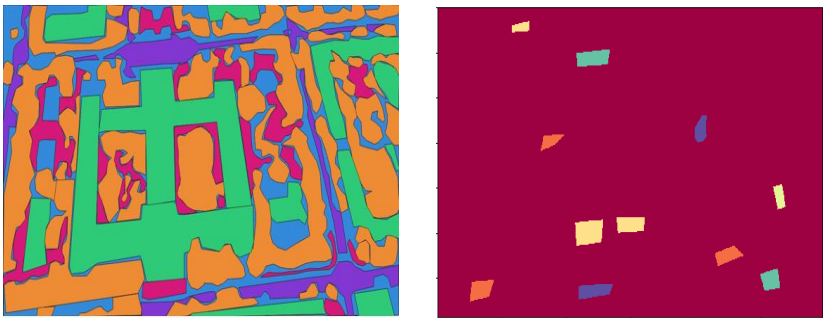
Берілген 2.25, 2.28 суреттерінде кескіннің гиперспектральды түрі көрсетілген, ал 2.26, 2.29 суреттерінде топтық шешімдер алгоритмі

келтірілген. 2.27, 2.30 суреттерінде салыстырмалы нәтижелер алу үшін XGBoost алгоритмінің нәтижесі келтірілген.



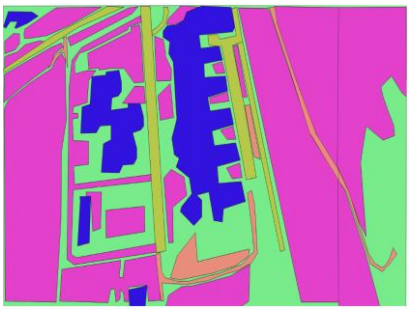
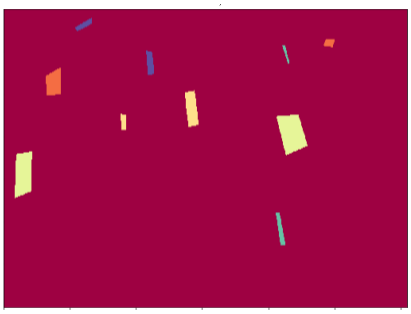
Сурет 2.30 – XGBoost алгоритмінің нәтижесі

Экспериментальды зерттеуде топтық шешімдер алгоритмдерін қолдана отырып жартылай бақылау арқылы оқыту есебінде деректердің бір бөлігі белгіленген деректер болып алынады, яғни гиперспектралді кескіндердегі әр кластың 1 %-нан тұратын белгіленген деректер ретінде аламыз. Белгіленген деректер бөлігі кездейсоқтықпен алынады. Келесі 2.31 және 2.32-суреттерде кескіндердің белгіленген объектілері берілген.



Сурет 2.31 – Ғылым ордасы кескіні (сол жақта топтық шешімдер алгоритмі үшін, оң жақта XGBoost алгоритмі үшін белгіленген объектілер)

Барлық каналдар бойынша пикселдердің сперктрлік жарығының мәнінен тұратын шулы деректер кестесі топтық шешімдер алгоритмінің кіріс деректері ретінде алынды, кластерлік алгоритмдердің топтық шешімінің базалық алгоритмі ретінде K-орта топтар алгоритмі алынды. Кластарға бөлудің әртүрлі нұсқалары [40, 40+M] интервалындағы кластерлер санын өзгерте отырып алынды, мұндағы *M=130*. Сонымен қатар, есептерді шешудің әр нұсқасын құру үшін кездейсоқ түрде каналдар алынды. Ұсынылған алгоритмді жұмыс орындалу барысында танымал, қазіргі таңда көп қолданысқа ие XGBoost топтық шешімдер алгоритмі және тірек векторларының әдісімен салыстыру жүргізілді.

Сурет 2.32 – *СДУ-і* кескінінің белгіленген объектілері (сол жақта топтық шешімдер алгоритмі үшін, оң жақта XGBoost алгоритмі үшін)

Бүгінгі таңда алгоритмдердің топтық шешімдерін қолданушы көптеген алгоритмдер бар. Солардың бірі Gradient Boosting Decision Tree (GBDT), XGBoost және pGBRT. Gradient Boosting Decision Tree (GBDT) алгоритмі машиналық оқытудың танымал алгоритмі болып табылады және көптеген классификацияларды орындауда XGBoost және pGBRT сияқты өте тиімді болып табылады. Дегенменде мұндай классификациялаулар көп инженерлік тиімділеу тәсілдері қолданылғанымен деректер көлемі үлкен, объект өлшемі жоғары болған жағдайда сапа көрсеткіші әлі де қағаттандырарлықтай емес. Бұл жердегі негізгі мәселе әр функция үшін мүмкін болатын барлық бөлулердегі объектілердің ақпараттылығын анықтауда объектілердің барлығын қарастырып отыру қажет болады, ал бұл өте күрделі жұмыс болып табылады. Ал осындай мәселелерді шешу үшін шет елдік ғалымдар Guolin Ke және т.б. градиенттік біржақты таңдама (GOSS – Gradient-based One-Side Sampling) және белгілерді эксклюзивті байланыс жасау (EFB – Exclusive Feature Bundling) екі жаңа әдісін ұсынды. Екі әдісті біріктіре отырып алынған алгоритмді LightGBM деп атады. LightGBM-і алгоритмі қарапайым GBDT алгоритміне қарағанда оқыту процесін жылдамдатады [62].

Ағаш түріне жататын бустинг алгоритмдердің бірі (GBDT) XGBoost алгоритмін Вашингтон университетінің ғалымдары Carlos Guestrin және Tianqi Chen еңбектерінен көруге болады. XGBoost - бұл соңғы жылдардағы ағаштар түріндегі градиентті бустинг алгоритмінің ең танымал және тиімді алгоритмдерінің бірі және ол scikit-learn сияқты кітапханалардың барлық мүмкіндіктерін қолдайды [63].

Төменде шулы параметрлердің кейбір мәндері үшін, яғни шу параметрлерін өзгерте отырып Ғылым ордасы кескінінің гиперспектральды бейнесінің белгіленбеген пикселдерін классификациялау дәлдігінің мәні көрсетілген. Алгоритм жұмысының уақыты оперативті жады көлемі 4 Гбайт, 2.8 ГГц тактілі жиіліктегі Intel Core i7 екі ядролы процессорында 2 минут шамасына созылды. Бұл кестеде топтық шешімдер алгоритмі XGBoost және тірек векторларының әдісіне негізделген алгоритмдеріне қарағанда жақсы нәтиже беретінін көреміз. Тірек векторларының әдісімен есептеуде жеке алгоритм болғандықтан аз уақыт кетті, дегенмен де дәлдік көрсеткіші шуды ұлғайтқан сайын төмендейтінін байқаймыз.

Кесте 2.3 – Шу параметрлерінің әртүрлі мәнінде алгоритмдер дәлдігі

Шу параметрі r	0%	10%,	20%	30%	есептеу уақыты
Топтық шешім алгоритмі	81,9%	79%	77,2%	74%	2 мин
XGBoost алгоритмі	81.1%	78.2%	73.6%	70.5%	4 мин 37 с
Тірек векторларының әдісі	82%	75%	71%	65%	1 мин 42 с

Бөлім бойынша қорытынды

Бұл бөлімде топтық шешімдер құрудың тәсілдері, бірнеше типтері қарастырылды, оларға анықтамалар мен түсініктер берілді. Оқыту ақпараттарының толық емес жағдайындағы классификациялау жағдайында қолданылатын тәсілдер қарастырылды. Осы қарастырылған тәсілдер негізінде гиперспектральды бейнелерге кластерлік талдау жасаушы алгоритмдер құрастырылды.

Бұл жерде негізгі мақсатымыз топтық кластерлік талдау алгоритмі мен классификациялаудың ядролық тәсілін үйлестіре отырып шешім алу, яғни біртекті топтық шешімдер құру тәсілі мен классификациялаудың ядролық тәсілдерін (*SVM*) үйлестіру жасалынды. Жұмыста алға қойылған есепті шешуде оқыту ақпараттары толық емес жағдайында бейнені тану есебі қарастырылды. Кластерлік талдаудың топтық шешімдер алгоритмдері, аз рангілі матрицалық декомпозиция және ұқсастық графының лапласианын реттеуге негізделген тәсілдердің комбинацияларын қолданумен берілген есепті шешу алгоритмі жасалынып, осы алгоритм арқылы алынған нәтижеге салыстырмалы талдаулар жүргізілді.

Жүргізілген эксперименттерден диссертациялық жұмысты орындау барысында алынған топтық шешімдер алгоритмдері коассоциациялық матрицаның азрангілі түрде қолданбайтын басқа топтық шешімдер алгоритмдеріне қарағанда дәлдігінің жоғарылығы көрсетілді.

3. АҚПАРАТТЫҚ ЖҮЙЕНІ ЖОБАЛАУ ЖӘНЕ ІСКЕ АСЫРУ

Ақпараттық жүйе – бұл ақпаратты өңдеуде, сақтауда және беруде пайдаланылатын құралдардың, әдістер мен тұлғалардың өзара байланысты жиынтығы. Ақпараттық жүйе әртүрлі салаларды қолданысқа ие. Атап айтсақ медицина саласында әртүрлі ауруларды классификациялауда [64], археология саласында әртүрлі қазбаларды классификациялауда, банк жүйесінде әртүрлі клиенттердің төлем қабілеттіліктерін ажыратуда. Яғни алдағы уақытта класс бойынша өңдеп талдау арқылы үлкен деректерді классификациялау тиімді және пайдалы болып табылады.

3.1 Ақпаратық жүйе құрылымы

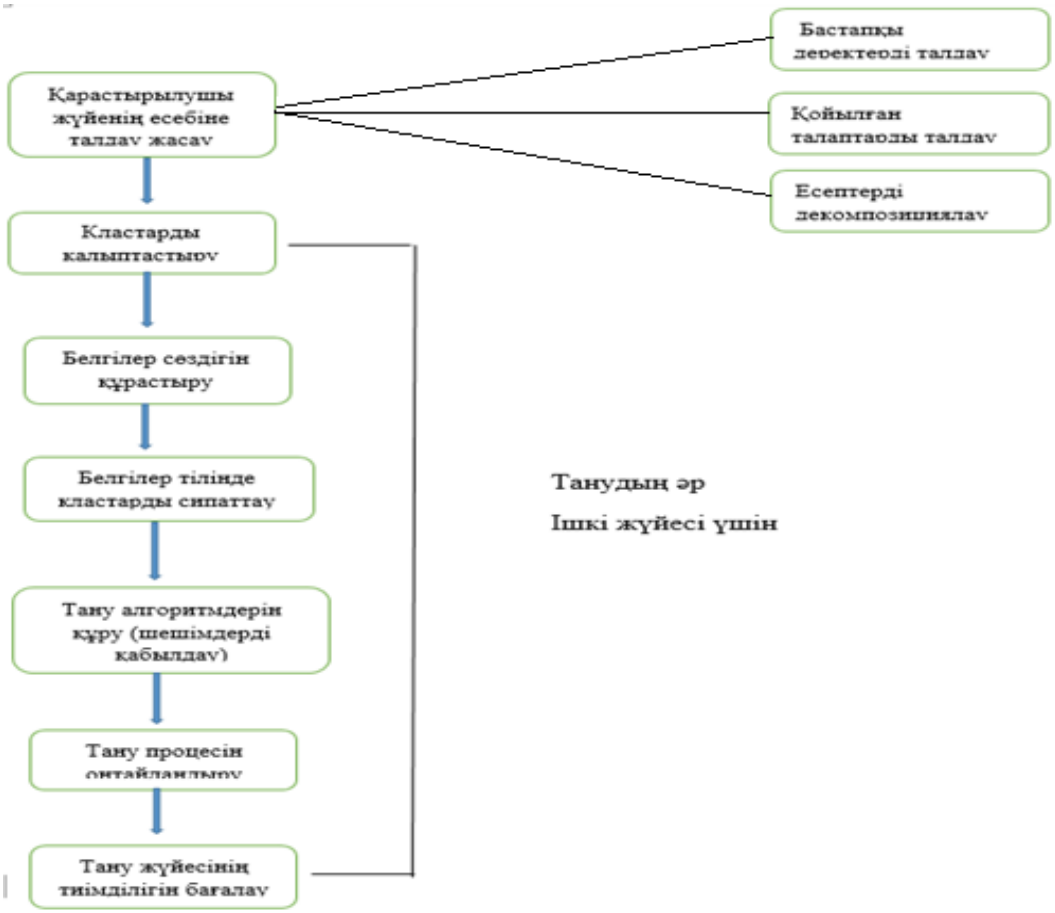
Бейнелердің белгі-сипаттамаларының физикалық табиғатына қарай тану жүйесі екі топқа бөлінеді: *қарапайым және күрделі.* Ал бастапқы ақпараттардың толықтығына қарай тану жүйесі оқытылатын, оқытусыз, өзін-өзі оқытушы жүйелер болып бөлінеді.

Оқытусыз жүйелерде танудың таңдап алынған принципі мен қолда бар ақпараты барлық қажетті кластарға қатесіз бөлуге мүмкіндік береді. Мұндай жүйеге мысалы ретінде эталондармен салыстыру принципіне негізделген жүйелерді алуға болады. Оқытусыз жүйелер әр кластың вектор-бейнелері қиылыспайтын шағын топтар түзетіндей кластеризациялаудың принципімен де құрылуы мүмкін.

Егер зерттеу объектісі туралы деректер оларды дәл бөлу үшін жеткіліксіз болса, оқытусыз тану жүйелері өте үлкен қателікке әкеледі. Керісінше, объектілер туралы ақпараттар саны қажет кластарды ерекшелеу үшін бұл деректерді толықтай қолданудың қажеті жоқ соншалықты үлкен болуы мүмкін. Мұндай жағдайда оқытылатын тану жүйелері қолданылады.

Көп жағдайда оқытатын жүйелер қасиеттер ортақтығы принципіне негізделген. Бұл жағдайда бейнелер міндетті түрде қандай да бір құрылыммен берілуі міндетті емес. Бұл мүмкін тек әр класта нақты бір анықталған статистикалық үлестірілу заңына бағынатын өлшем-параметрлер жиыны немесе нақты бір комбинациямен әр класта бар бөлінбейтін қандай да бір элементтер жиыны болуы мүмкін. Мұндай жиындарды бейнелердің жалпыланған теориясында еркін конфигурациялау деп атайды. Оқытудың міндетті әр кластар үшін бейне қасиеттерін немесе олардың арасындағы өзара байланысты анықтау. Кластеризациялау принципін қолданушы оқыту жүйесі тану процесінде қателіктің мөлшерін мүмкіндігінше азайумен қамтамасыз ететін барлық вектор-бейнелерді кластерлерге бөлуді іздеуден тұрады.

Өзін-өзі оқытушы жүйелерде оқыту процедурасы тану процесінің өзінде арнайы алгоритммен орындалады. Танудың сапасын бағалау ретінде оқыту процесінде максимизацияланатын немесе минимизацияланатын тану қателіктерімен байланысты қандай да бір функционал қолданылады.



Сурет 3.1 – Тану жүйесін құрудың жалпы схемасы

Ақпараттық жүйедегі қолданушы интерфейстер

Қолданушының графикалық интерфейсі келесі әрекеттерді орындау керек:

- Әртүрлі дерекқорларды сақтаушыда объектілерді сақтау;
- Бастапқы мәліметтерді түзету мен көру;
- Нәтижелерді қарастыру;
- Бір ішкі жүйенің бірнеше алгоритмдерімен біруақытта тізбектей қолданушының бір ғана командасымен деректерге өңдеу жүргізу керек;
- Көп есепті режимде параллель біруақытта деректерге өңдеу жүргізу;
- Қолданушының талабы бойынша кез-келген уақытта деректерді өңдеуді тоқтату керек;
- Алгоритм туралы есеп беру құру.

Ақпараттық жүйенің программалық түрде жүзеге асырылуы

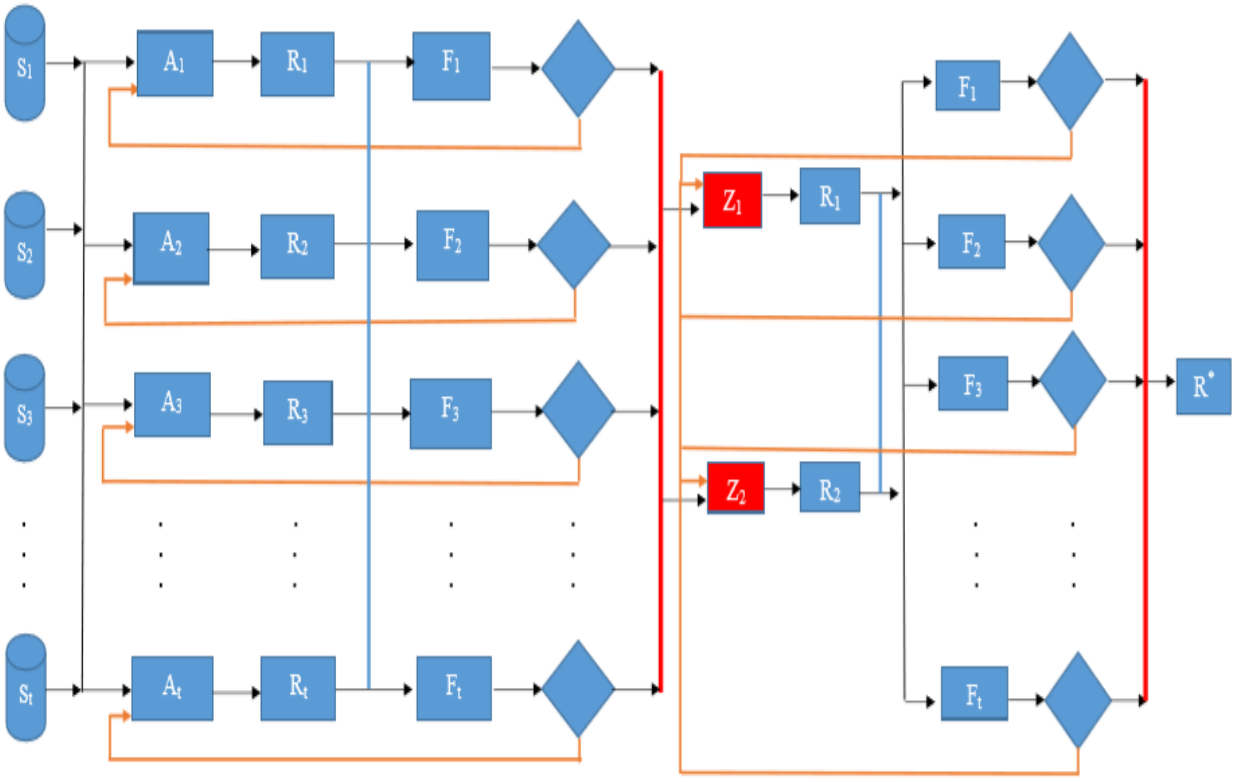
Берілген диссертациялық жұмыста бейнені тану мен кластеризациялаудың ақпараттық жүйесі (АЖ) жасалынды. Бұл жүйе арқылы қандай да бір объектілер жиынын кластерлік талдау алгоритмдері арқылы нәтижелік бөлуге болады. Ақпараттық жүйелерде жасауда келесі алдыңғы қатарлы программалық технология UML моделдеу тілі қолданылды.

Ақпараттық жүйелерде ақпараттарды алдын ала өңдеу жүргізіледі. Ол үшін белгілерді іріктеп алу критериі ретінде энтропия түріндегі функция қолданылды. Энтропия анықталмағандықтың статистикалық өлшемін көрсетеді:

$$H = - E_p\{ln\ p\}$$

мұндағы, p – бейнелер жиынтығы ықтималдығының тығыздығы, E_p – p тығыздығының математикалық күтімінің операторы. Энтропия түсінігін белгілердің ақпараттық жиынтығын ұйымдастырудағы критериі ретінде қолдануға болады. Жүйеде қолданылатын деректер объектілер жиынының сипатталуы болып табылады. Ал ішкі жүйелердің деректері басқа ішкі жүйелер жұмысының нәтижелері болуы да мүмкін.

Жүйе жұмысының нәтижесі объектілер кластарының жиыны болып табылады. Программалық жүйе жаңа алгоритмдер мен ішкі жүйелерді толықтырып отыруға ыңғайланып жасалынуы керек. Ол үшін жалпы жүйенің толық кодына минималды түрде өзгертулерді қажет етуші жаңа алгоритмдер мен ішкі жүйелерді қосып отыруға ыңғайлы интерфейстер жиынын құру керек. Төмендегі 38-суретте ақпараттық жүйелерде құрылымдық схемасы берілген. Құрылымдық схема екі этаптан тұрады.



Сурет 3.2 – Ақпараттық жүйенің құрылымдық схемасы

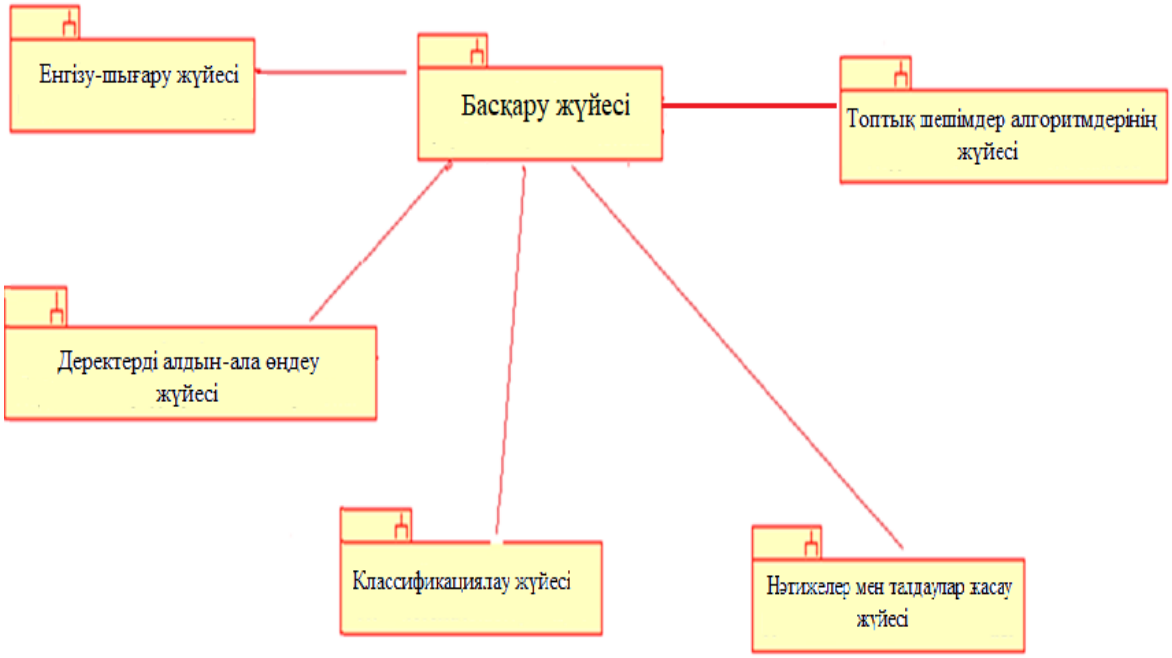
Ақпараттық жүйенің құрылымдық схемасы ақпараттық жүйенің топтық шешімдер алгоритмдері және жеке алгоритмдер арқылы орындалу процесін сипаттайды. Мұндағы $S_1,\ldots,S_t$ бастапқы объектілер жиыны. Осы объектісіне $A_1,\ldots,A_t$ алгоритмдерін қолдана отырып $R_1,\ldots,R_t$ шешімдерін аламыз. Сапа көрсеткіштерін анықтаудың $F_1,\ldots, F_t$ функционалдарын пайдаланып, әр шешімнің көрсеткішін анықтаймыз да, қойылған талаптарды, шарттарды тексеріп, егер шарт қанағаттандырған жағдайда келесі этапқа жіберіледі, кері жағдайда басынан басталады. Функционалдар түрі өте көп, олардың бірнеше түрлері төменде келтірілді. Екінші этапта қойылған талаптарды қанағаттандырған шешімдер диссертациялық жұмыста қойылған орталық объектілерді оқшаулау және жартылай бақылау арқылы оқыту есебінің топтық шешімдер алгоритмдерін құруда қолданылады. Бұл жерде қойылатын шарт қолданушының немесе тапсырыс берушінің талабы бойынша қойылады. Есептеу барысында қолданылған алгоритмнің нәтижесі шартты қанағаттандырмаған жағдайда қайта бастапқы қадамға барады. Осылайша бірнеше итерация орындалады.

Ақпараттық жүйе мынандай ішкі жүйелерден тұрады:

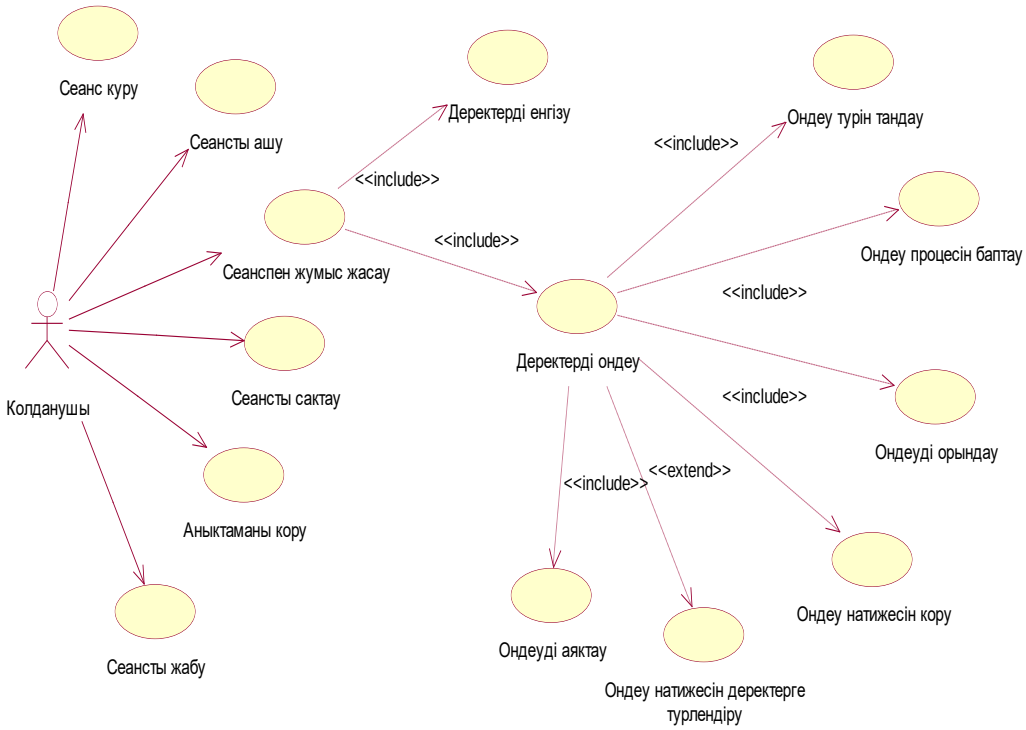
- басқарушы жүйе;
- деректерді енгізу-шығару жүйесі;
- деректерді алдын-ала өңдеу жүйесі;
- топтық шешім қабылдау жүйесі;
- нәтижелерді талдау және бағалау жүйесі;

Барлық деректерді өңдеуші ішкі жүйелер бір немесе бірнеше алгоритмдерден тұрады.

Басқарушы ішкі жүйе барлық ішкі жүйелерді бақылап отырады. Осы сияқты АЖ-нің барлық ішкі жүйелерінің атқаратын қызметтері бар. Ақпараттылықты көрсету деректердегі ақпараттардың айырмашылық дәрежесі мен оның компоненттерінің ақпараттылығына байланысты [65,66]. Логикалық формулаларда ақпараттылықтың өлшемін шығару есебі мен оларды классификациялау осы жиында ара қашықтықтарды анықтау мәселесін ұсынады. Құрылған ақпараттық жүйелердің басқару жүйесі келесі ішкі жүйелерден тұрады:



Сурет 3.3 – Ішкі жүйелер диаграммасы



Сурет 3.4 – Прецеденттер диаграммасы

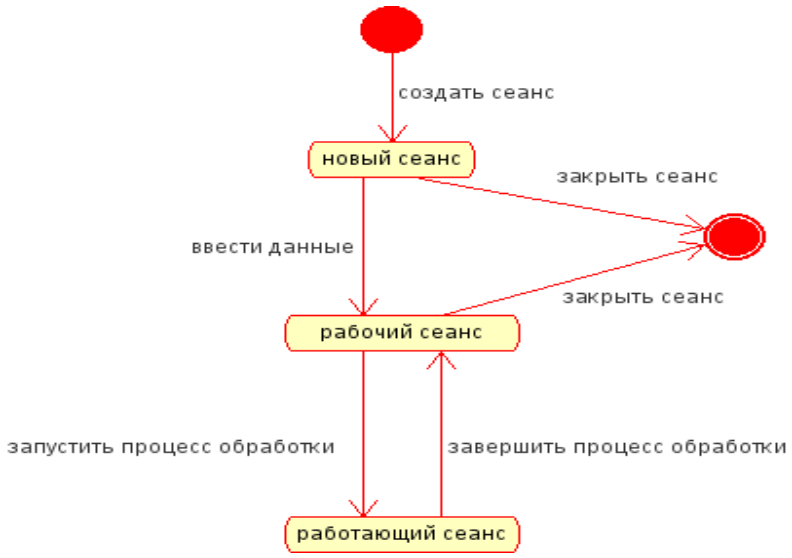
Нәтижелерді бағалау мен талдау ішкі ақпараттық жүйесі кластерлік талдаудың әртүрлі нәтижелерінің көрсеткіштерін есептейді. Сапасын бағалау критериі ретінде объектілердің үлестірілулері, объектілердің центрден орташа ауытқуларын, т.б. анықтайтын функциялар алынады.

Нәтижелерді көрсету ішкі жүйесі есеп берулерді құру үшін қолданылады.

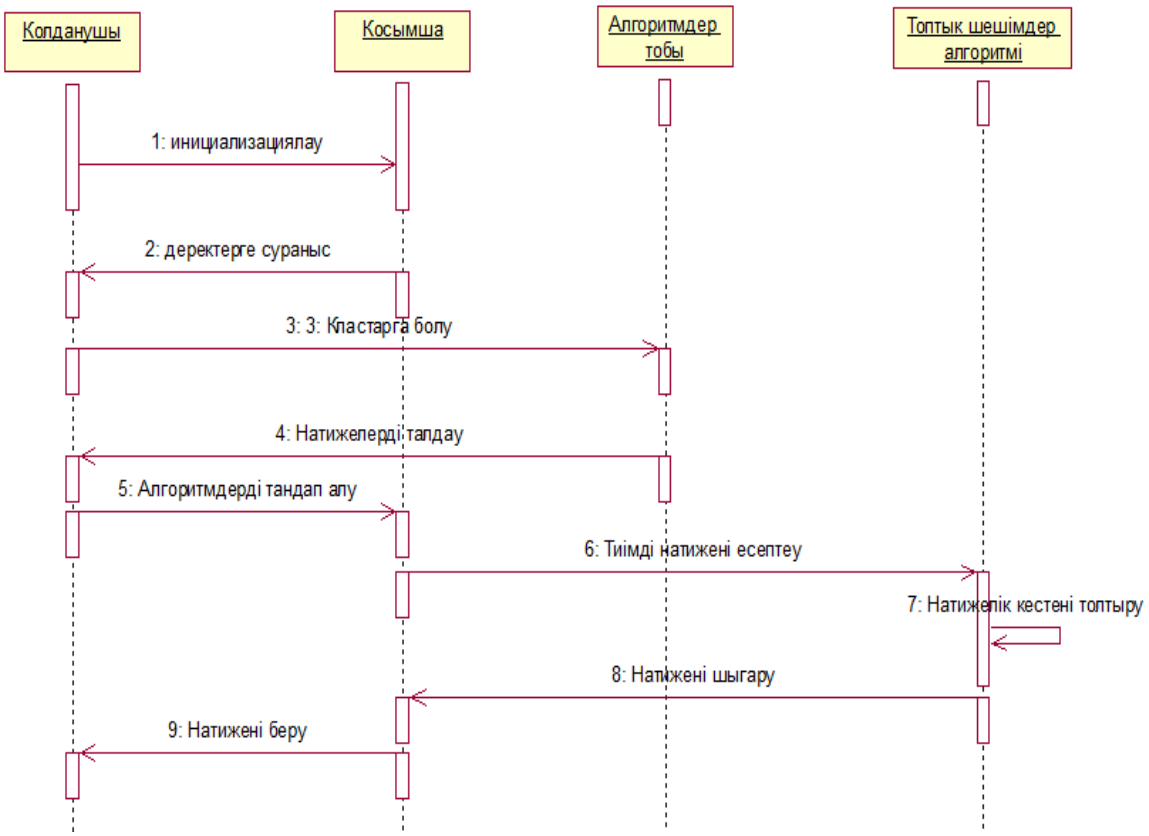
Қолданушының интерфейсінің ішкі жүйесі жүйенің барлық функцияларын қолданудың қарапайым және ыңғайлы мүмкіндігін береді. Ары қарай 3.3-суретте ішкі жүйелер диаграммасы көрсетілген.

3.2 Күй диаграммасы. Жүйенің күйін сипаттау

Күйі	Күйді сипаттауы
Жаңа сеанс	Қолданушы деректерді енгізуге, қарауға, түзетуге және сақтауға мүмкіндігі бар.
Жұмыс сеансы	Қолданушы деректерді енгізеді және деректермен жұмыс жасап, түзеп, талдаулар жасай алады.
Жұмыс жасаушы сеанс	Қолданушы деректерді өңдеу процесін іске қосты және деректер мен алынған нәтижелерге ешқандай өзгерістер жасай алмайды, дегенменде жүйені жұмыс сеансы күйіне ауыстыра отырып өңдеу процесін тоқтата алады.



Сурет 3.5 – Сеанс күйінің диаграммасы



Сурет 3.6 – Топтық шешімдер ішкі жүйесінің тізбектелу диаграммасы

3.3 Ақпараттық ішкі жүйелер және олардың программалық түрде жүзеге асырылуы

Берілген диссертациялық жұмыста тану және кластеризациялаудың программалық жүйесі берілген және ол Java, Python және Delphi программалау тілдерінде программа коды жазылған.

Деректерді алдын-ала өңдеу ішкі жүйесі

Бастапқы мәліметтер жиынтығын қалпына келтіру операциялары, сонымен қатар бөлуді орындауда қолданылған белгілердің ақпараттылығын тексеру жасау.

Бейне тану алгоритмдерінің ақпараттық ішкі жүйесі

Диссертациялық жұмыста құрылған ақпараттық жүйеде топтық шешімдер алгоритмін құруда келесі базалық алгоритмдер қолданылды:

A_1: k ішкі топтар алгоритмі, кластерлер саны – 5;
A_2: k ішкі топтар алгоритмі, кластерлер саны – 6;
A_3: MaxMin (кездейсоқ бірінші центр) алгоритмі + ЖК;
A_4: MaxMin (центрлері екі алшақ нүктелер алынған) алгоритмі + ЖК;
A_5: MaxMin (кездейсоқ центр) алгоритмі + k means (кластарды 5 дейін қысқарта отырып);
A_6: MaxMin (кездейсоқ центр) алгоритмі + k means (кластарды 4 дейін қысқарта отырып);
A_7: MaxMin (кездейсоқ бірінші центр) алгоритмі + k means;

A_8: MaxMin алгоритмі + k means (шектеуді 0,3 дейін түсіру);

A_9: MaxMin алгоритмі + k means (шектеуді 0,3 дейін түсіру);

A_{10}: MaxMin (кездейсоқ бірінші центр) алгоритмі + БЕА(бағаларды есептеу алгоритмі).

Бұл алгоритмдер Евклид метрикасына, эталондарды қысқартуға негізделген.

Жүйедегі ішкі жүйе «топтық шешімдердің ішкі жүйесінде» орталық объектілерді ерекшелеу арқылы және жартылай бақылау арқылы оқыту алгоритмдері жүзеге асырылды.

Ал «нәтижелерді бағалау және талдау ішкі жүйесінде» нәтижелерді талдау мен бағалау жүргізілді. Ішкі жүйесі келесі кластардан тұрады:

AnalysisSubsystem класы – ішкі жүйенің негізгі класы:

поле ai – алгоритм параметрлері;

поле report – бағалар нәтижелерінің есебі;

getReport() – бағалар нәтижелерінің есебін қайтарады;

Класс AAnalysis – ішкі жүйелер алгоритмдерінің негізгі өрістері мен тәсілдерін көрсетуші абстрактілі класс.

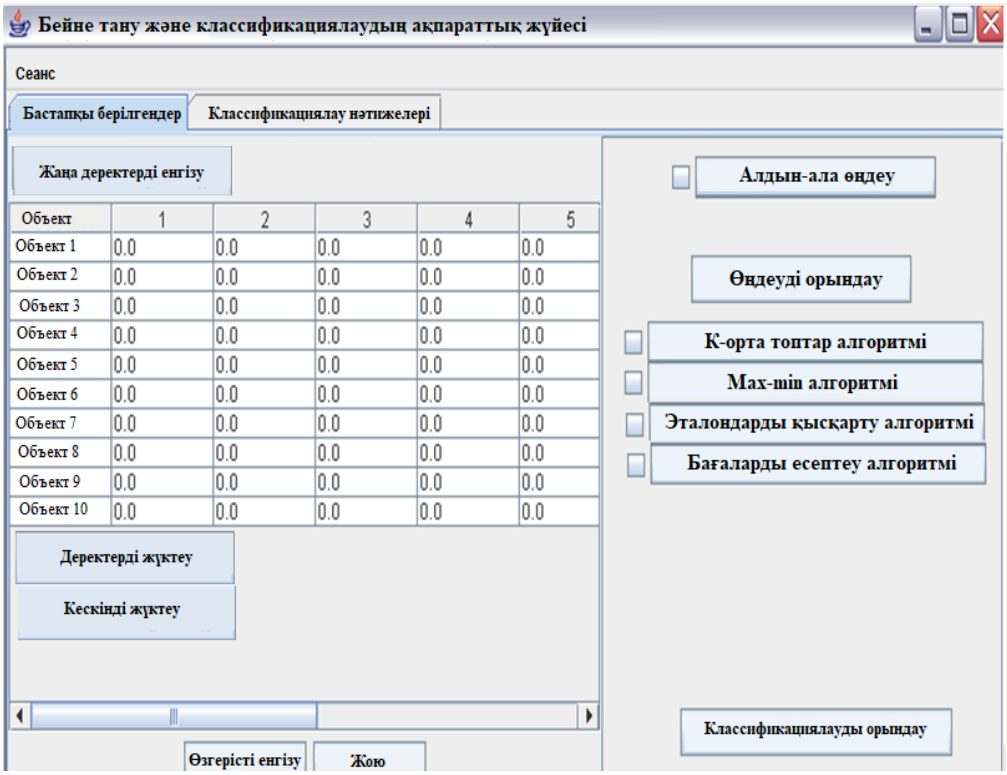
Analysis1, ... Analysis6 – нақты алгоритмдер үшін AAnalysis-ті нақтылаушы кластар.

Нәтижелерді талдау барысында келесі нәтижелер есептеледі:

- кластар орталықтарынан ауытқу;
- класс ішілік қашықтық;
- кластар арасындағы орташа қашықтық;
- класс ішілік қашықтықтардың қосындысы;
- жинақтылық;
- объектілер санының бірдейлігі
- алшақтық;

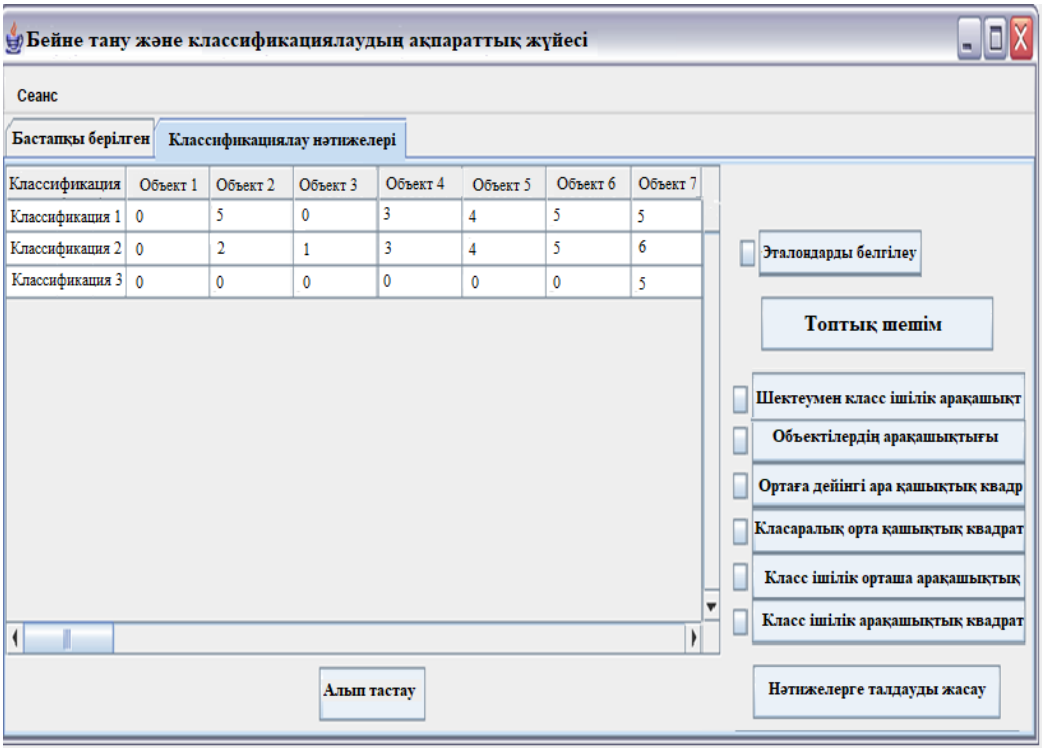
Жүйенің программалық құрылымы

Құрылған программалық жүйені іске қосуда келесі терезе іске қосылады.



Сурет 3.7 – Жүйенің алғашқы беті

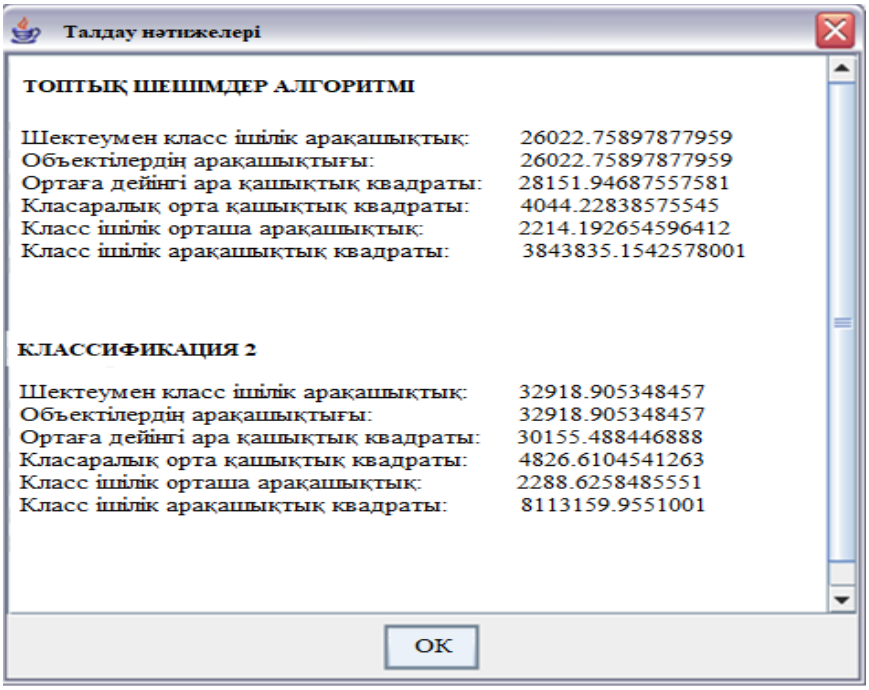
Берілген объектілерге басты терезенің оң жағында тұрған «белгілемелерді орната отырып алгоритмді таңдаумен классификациялауды жасауға болады. Бұда кейін «Классификациялар нәтижелері» терезесінде таңдап алынған алгоритм жұмысының нәтижесі пайда болады (3.8 – сурет).



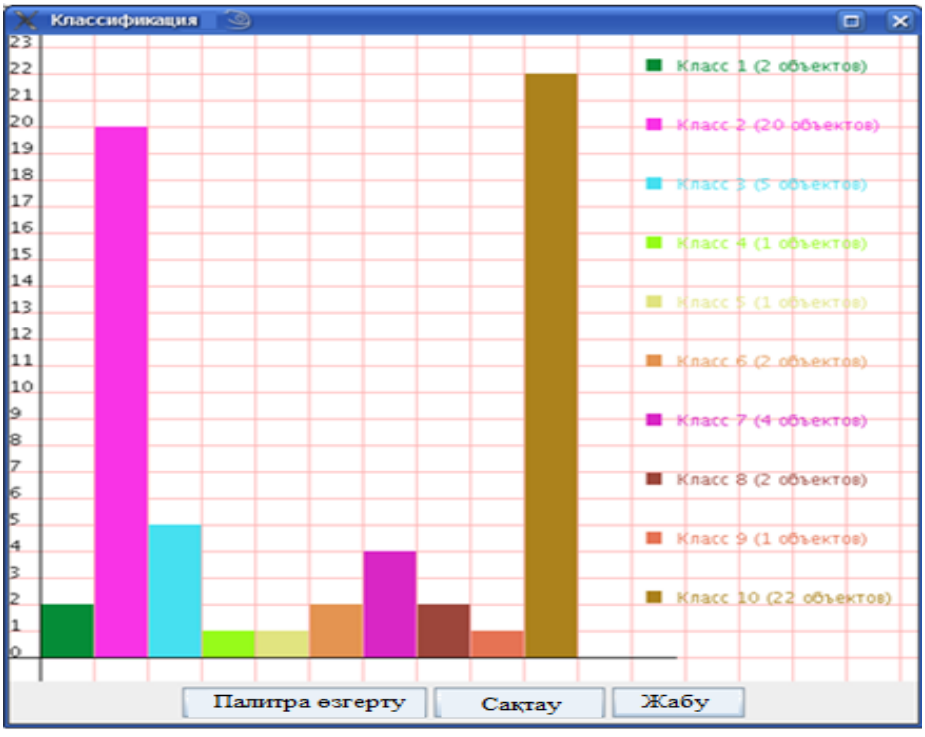
Сурет 3.8 – K орта топтар алгоритмінің нәтижесі

3.4 АЖ қолданылған алгоритмдердің нәтижелерін бағалау

Бағалау функциясы түріндегі функционал алгоритмдердің жұмысына талдау және бағалау жүргізеді. Функционалдардың әр түрі бар және олар тұтынушының талабына баланысты қолданылады. Нәтижелерге талдау жасауда алынған бағалау нәтижесі келесі суретте берілген, ал сапасын бағалау функциналдары келесі бөлімде сипатталған.



Сурет 3.9 – Ақпараттық жүйелердегі бағалау нәтижелері



Сурет 3.10 – Бағалау нәтижелерін визуализациялау

Алынған нәтижелерге талдау жүргізу

Деректерді кластерлік талдау – алгоритмдер мен моделдері айтарлықтай көп, есептерді шешудің бірнеше тәсілдері бар динамикалық түрде дамушы аймақ. Кластерлеу есептерін шешу барысында олардың әр этаптарында қорытынды нәтижеге әсер етуші жағдайлар болуы мүмкін. Бұл жағдайлар кластерлеудің әр этаптарында келесі себептермен пайда болуы мүмкін:

1) бастапқы объектілер сипаттамаларын қате беру (текстерді векторлық түрде беру жағдайы жиі орын алады);

2) объектілерді салыстыру үшін анықталған өлшемдер дұрыс емес;

3) бастапқы деректерге қолданылатын кластерлеу алгоритмінің өзін дұрыс таңдамау.

Қазіргі таңда әртүрлі функцияларды қолдануға негізделген сапаны бағалаудың әртүрлі танымал тәсілдері бар [67]. Кластерлеу сапасын бағалау үшін әртүрлі кластерлек сапа индекстері де қолданылады.

Айталық, бастапқы берілген объектілер жиыны $X = \{x_1, x_2, \ldots, x_n\}$ және $S_1, \ldots, S_r$ олардың қандай да бір кластерлеу алгоритмі арқылы алынған бірнеше бөлінулері болсын. $S_1, \ldots, S_r$ бөлінулер жиыны негізінде S қорытынды бөлінуі құрылады. Кластерлік талдаудың классикалық есебінде осы бөлінулер тиімділеудің кейбір $F(X,S)$ критериін қанағаттардыру керек. Тәжірибеде көп жағдайда қолданылатын критериі

$$F(X,S) = \sum_{j=1}^{c} \sum_{x \epsilon S_j} \left\| x - \bar{x}(S_j) \right\| \rightarrow min, \qquad (3.2)$$

мұндағы $\bar{x}(S_j) = \frac{1}{|S_j|} \sum_{x \epsilon S_j} x$ – j-ші кластердің центроиды.

Кластерлеу сапа индекстері кластерлеу нәтижесін нақты бөлу нәтижелерімен салыстыратын және берілгендердің жақындық өлшемін анықтаудың әртүрлі тәсілдерін қолданатын деп екіге бөлінеді.

Бірінші типке жататын индекстердің бірі Ранда индексі:

$$RI = \frac{A+\mathrm{B}}{C_N^2},$$

мұндағы A – бірдей кластерлерге кіретін объектілер жұбының саны, B – әртүрлі топтарға кіретін жұптар саны.

Енді топтаудың сапа индексінің екінші типі «ішкі» индекстерді қарастырайық. Алдын ала келесідей белгілеу енгізейік: Q_0 – арқылы бір кластерге тиесілі объектілер жиынының санын белгілейік ($Q_0 = \sum_k C_{N_k}^2$); $\bar{X}_k$ - арқылы k-шы кластердің орта мәнінің (центроид) векторын; $\overline{\mathrm{X}}$ – арқылы барлық бақылаулар үшін орта мәндер векторын; $\sum \min(n)$ арқылы қосылғыштардың n ең кіші мәндерінің қосындысын, ал $\sum max(n)$ – арқылы n ең үлкен мәндерінің қосындысын белгілейік.

Бірнеше кластерлеудің сапа индекстерін келтірейік; мұнда әр индекс үшін минимизациялау (↓) әлде максимизациялау (↑) қажеттілігі көрсетіледі.

Дунна индексі (↑):

$$D(G) = \frac{\min\limits_{C^{(k)}\epsilon G}\{\min\limits_{C^{(l)}\epsilon G/C^{(k)}}\{\delta(C^{(k)},C^{(l)})\}\}}{\max\limits_{C^{(k)}\epsilon G}\{\triangle(C^{(k)})\}},$$

мұндағы $\triangle(C^{(k)},C^{(l)}) = \min\limits_{o^{(i)}\epsilon C^{(k)}}\min\limits_{o^{(j)}\epsilon C^{(l)}}\{\rho(x^{(i)},x^{(j)})\}$,

$$\triangle\left(C^{(k)}\right) = \max\limits_{o^{(i)},o^{(j)}\epsilon C^{(k)}}\{\rho(o^{(i)},o^{(j)})\}.$$

Калински-Харабаша индексі (Calinski-Harabash) (↑):

$$CH(G) = \frac{N-K}{K-1}\frac{\sum_{C^k\epsilon G} N_k\rho(\overline{X_k},\bar{X})}{\sum_{C^k\epsilon G}\sum_{o^{(j)}\epsilon C^k}\rho(X(o^i),\bar{X}_k)}.$$

Бұл жерде кластерлердің алшақтығы жалпы центроидтар және кластерлер центроидтары арасындағы ара қашықтықтар негізінде есептелінеді.

Гамма индексі (↓):

$$\Gamma(G) = \frac{\sum_k\sum_{o^{(i)},o^{(j)}\epsilon C^k} dl(o^{(i)},o^{(j)})}{Q(C_N^2-Q_0)},$$

мұндағы $dl(o^{(i)},o^{(j)})$ келесі төмендегідей шарттарды қанағаттандыратын $o^{(l)},o^{(m)}$ деректері объектілеріндей мүмкін болатын жұптарды білдіреді:

а) $o^{(l)}$ және $o^{(m)}$ әртүрлі кластерлерге тиесілі,

б) $\rho(o^{(l)},o^{(m)}) < \rho(o^{(i)},o^{(j)})$.

С-индексі (↓):

$$C(G) = \frac{S(G)-S_{min}(G)}{S_{max}(G)-S_{min}(G)}$$

мұндағы

$$S(G) = \sum_k\sum_{o^{(i)},o^{(j)}\epsilon C^{(k)}}\rho\,(o^{(i)},o^{(j)})$$

$$S_{min}(G) = \sum_{o^{(i)},o^{(j)}}\min(Q_0)\{\rho(o^{(i)},o^{(j)})\}$$

$$S_{max}(G) = \sum_{o^{(i)},o^{(j)}}\max(Q_0)\{\rho(o^{(i)},o^{(j)})\}$$

Әртүрлі әдістерді пайдалану арқылы ұқсас кластерлердің құрылуы кластеризациялаудың дұрыстығын білдіреді. Бір ғана объектілер жиынына әртүрлі тәсілдерді қолдануда қорытынды бөлудің бірнеше нұсқаларын алуымыз мүмкін. Өйткені кластерлік құрылым сипаттамасына біріншіден классификациялау жүргізілетін белгілер жиыны, екіншіден таңдап алынған алгоритм типі әсер етуі мүмкін. Мысалы, иерархиялық және итеративті әдістер кластерлердің санының әртүрлі болуына әкеледі. Сондықтан кластерлердің өзін құрамы, объектілердің жақындық дәрежесі бойынша анықтайды. Бұл жерде классификациялау тәсілдерінің «ең жақсысын» таңдау деген мәселе туындайды. Кластеризациялау мен тану есебінде объектілер жиынын кластерлерге бөлуші тәсілдердің сан алуан түрлері бар екені белгілі. Бұл тәсілдер сапасының салыстырмалы түрде талдау жасау қажеттілігі туындайды. Осы қажеттілікті қанағаттандыру мақсатында бөлудің сапа функционалы қолданылады, оны *Q(S)*-деп белгілейік. Кластеризациялау есебін дискретті оптимизациялау есебі сияқты қоюға болады: x_i объектілеріне y_i кластерлер номерлерін таңдап алынған сапа функционалының мәні дұрыс мән қабылдайтындай жазу керек. Сапа функционалдарының көптеген түрлері бар, бірақ олардың ішінде біреуін «ең жақсысы» деп айту қиын.

Ең алғаш 1965 жылы нақты оқытусыз статистикалық және оптимизациялау есебі мен жалпы түрдегі классификациялаудың сапа функционалы ұсынылған М.И. Шлезингердің [68] мақаласы жарық көрді. Бұдан кейін көп ұзамай оқытусыз тану, яғни танудың маңызды теоретикалық құрылымының негізі қойылған көптеген жұмыстар жарық көре бастады. Бұл жұмыстарда кластеризациялау есебі үшін маңыздырақ екі бағыт: М.А. Айзерман, А.Г. Аркадьев, Э.М. Браверман, және т.б. [69] жұмыстарында зерттелініп, дамытылған потенциалды функция теориясы және Г.К.Кельманс, Я.З.Цыпкин [70] қарастырған тану есептеріндегі стохастикалық аппроксимация теориясы.

Осы және басқа да жұмыстарда жалпы сапа критерилерінен басқа осы күнге дейін мағынасын жоғалтпаған оптимизациялаудың әмбебап алгоритмдері де қарастырылған. Бұдан кейінгі жылдары осы бағытты дамыту қарқыны күшейе түсті: жаңа сапа функционалдары мен оптимизациялаудың жаңа алгоритмдері жасалына бастады.

Жалпы жоғарыда аталған жұмыстарда сапа критерилері ғана емес олардың сәйкес алгоритмдері де келтірілген. Айталық, d-классификациялау кеңістігіндегі X метрикасы және $S = (S_1, S_2, S_3, \dots, S_p)$ кейбір $(X_1, X_2, X_3, \dots, X_n)$ белгіленген объектілерін берілген p саны бойынша $S_1, S_2, S_3, \dots, S_p$ кластарына бөлу болсын. Онда төменде сипаттамалары қарастырылатын бірнеше функционалдарды келтіре кетейік:

1. Кластар центрлерінен ауытқу

$$F_1 = \sum_{i=1}^{k} \sum_{x_i \in S_l}^{m} (x_i - \bar{x}_l)^2$$

2. Ішкі кластар ара қашықтықтарының квадраты

$$F_2 = \sum_{l=1}^{k} \sum_{i,j \in S_l}^{m} d_{i,j}^2$$

3. Класс центрлеріне дейінгі (немесе ара қашықтықтар қосындысы) ара қашықтықтар қосындысы

$$F_3 = \sum_{l=1}^{k} \sum_{i \in S_l} d^2(x_i, x_l)$$

Функционалдың мұндай түрі өте кең таралған. Класс центріне дейінгі ара қашықтықтар қосындысы кластерлік талдау есептерінде қолданылып, жақсы нәтижелер алынуда.

4. Кластар арасындағы максималды ара қашықтық

$$F_4 = \min_{l,q}(\max_{l,q} \rho_{l,q})$$

Класс аралық минималды арақашықтықты максимизациялаудың алдыңғы функционалдардан айырмашылығы бөлудің параметрлік емес сипаттамасына негізделген. Функционал қарапайымдылығымен танымал. Кей жұмыстарды кластар ара қашықтығы жақын көрші принципімен анықталады.

Әр объектілер жиынына сәйкес сапа көрсеткішін анықтау керек. Жалпы кластерлеу саласында әмбебап тәсіл жоқ деп айтса болады. Жақсы сапа көрсеткішін алу үшін сапасын анықтауды тиімді түрде таңдау өте маңызды.

ҚОРЫТЫНДЫ

Диссертациялық жұмысты орындау барысында бейне тану және классификациялаудың заманауи теориясы мен практикасы қолданылып, келесідей нәтижелер алынды:

1. Классификациялау және тану есептерінде топтық шешімдер алгоритмдерін жалпы жан-жақты шолулар мен зерттеулер жүргізілді;
2. Топтық тану мәселелерінің қойылуында жартылай бақылау арқылы оқыту есебін зерттей отырып шешілді;
3. Классификациялау және тану есептерінде жаңа топтық шешімдер алгоритмдері құрылды;
4. Алынған нәтижелерге талдаулар жасалынды. Сапасына бағалаулар жүргізілді. Нәтижелер танымал классификациялау және бейне тану алгоритмдерімен салыстырулар нәтижелері көрсетіліп, алынған нәтижелер негізінде классификациялау және бейне танудың ақпараттық жүйесі құрылды.

ПАЙДАЛАНҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Гурянова (Кибитова) В.Н. Ансамбль алгоритмов для определения ишемической болезни сердца // Сборник тезисов XXVI международный научной конференции Ломоносов. – М: Издательский отдел факультета вычислительной математики и кибернетики МГУ имени М.В. Ломоносова, 2017. – С. 15-17.

2 «Цифрлық Қазақстан» мемлекеттік бағдарламасы // https://primeminister.kz/kz/page/view/tsifrlik_kazakstan_memlekettik_bagdarlamasi (алынған күні 13.05.2019).

3 Гусева А.И., Киреев В.С., Кузнецов И.А., Бочкарёв П.В. Исследование алгоритмов многомерной классификации научных данных // Фундаментальные исследования. – 2015. – № 11-5. – С. 868-874.

4 Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. – М.: Наука, 1974. – 418 с.

5 Загоруйко Н.Г. Прикладные методы анализа данных и знаний. – Новосибирск: Изд-во Института математики, 1999. – 41 с.

6 Амиргалиев Е.Н. Теория распознавания образов и кластерного анализа; М-во образования науки РК, КазНТУ им. К. И. Сатпаева. – Алматы: Изд-во КазНТУ, 2003. – 6 с.

7 Д.С. Черезов, Н. А. Тюкачев. Обзор основных методов классификаций и кластеризации данных // Вестник ВГУ, Серия Системный анализ и технологии. – 2009. – № 2. – С. 19-25.

8 У.Гренандер. Лекции по теории образов. – М.:Анализ образов. Мир.: 1981. – С. 180-189.

9 Zahn C. T. Graph-theoretical methods for detecting and describing gestalt clusters // IEEE Trans. Comput. – 1971. – P. 68–86.

10 Rand W. Objective criteria for the evaluation of clustering methods // Journal of American Statistical Association. – 1971. – V.66. – P. 846-850.

11 Городецкий В. И., Серебряков С. В. Методы и алгоритмы коллективного распознавания // Труды СПИИРАН, — СПб.: Наука, 2006. – С. 29-35.

12 В. В. Рязанов. Комитетный синтез алгоритмов распознавания и классификации // Ж. вычисл. матем. и матем. физ. U.S.S.R. Comput. Math. Math. Phys. – 1981. –Т. 21:6, – С. 172–182.

13 Ghosh J., Acharya A. Cluster ensembles // Wiley Inter disciplinary Reviews: Data Mining and Knowledge Discovery. – 2011. – Vol. 1(4). – P. 305-315.

14 Gowda, K. C., Krishna, G. Agglomerative clustering using the concept of mutual nearest neighborhood // Pattern Recognition. – 1977. – V. 10. – P. 105–112.

15 В.Б.Бериков. Коллектив алгоритмов с весами в кластерном анализе разнородных данных. // Вестник Томского Государсвенного Университета. – 2013. – № 2(23). – С. 22-31.

16 Masoudnia Saeed, Ebrahimpour Reza. Mixture of experts: A literature survey // Artificial Intelligence Review. –2014. – V. 42. 10. – 275-293

17 R. M.O., Cruz, Robert Sabourin, George D.C. Cavalcanti. Dynamic classifier selection: Resent advances and perspectives // Information Fusion. – 2018. – № 41. – P. 195–216.

18 G.Beliakov, T.Calvo, S.James. Consensus measures constructed from aggregation functions and fuzzyimplications // Knowledge-Based Systems. – V. 55. – 2014. – P. 1-8.

19 Y.Zhang, H.Zhang, J.Cai. A Weighted Voting Classifier Based on Differential Evolution // Abstract and Applied Analysis. – 2014. – V:1-6. – P.15-22.

20 Rafael M.O. Cruz, Robert Sabourin, George D.C. Cavalcant, Dynamic classifier selection: Recent advances and perspectives// Information Fusion. – 2018. – V. 41. – P. 195-216.

21 Воробьев Н. Н. Вопросы математизации принятия решений на основе экспертных оценок // Материалы IV симпозиума по кибернетике. – 1972. –Т. 3. – С. 47–51.

22 Rafael M.O. Cruz, Robert Sabourin, George D.C. Cavalcant, Dynamic classifier selection: Recent advances and perspectives // Information Fusion. – 2018. – V. 41. – P. 195-216.

23 Vega-Pons S. Weighted cluster ensemble using a kernel consensus function // Progress in Pattern Recognition Image Analysis and Applications. – 2008. – V. 5197. – 195-202.

24 Robert E. Schapire. Theoretical views of boosting and applications // Algorithmic Learning Theory, 10th International Conference, ALT ’99. – Tokyo, 1999.

25 Журавлёв, Ю. И. Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики. –1978. – Вып.33. — С. 5–68.

26 И.А.Кузнецов, В.С.Киреев // Труды XVIII Международной конференции DAMDID/RCDL’2016 «Аналитика и управление данными в областях с интенсивным использованием данных», Ершово, 11-14 октября 2016. [Електронний ресурс] // http://ceur-ws.org/Vol-1752/paper07.pdf. 15.10.2019.

27 Hongshan X iao, Zhi Xiao, Yu Wang. Ensemble classification based on supervised clustering for credit scoring // Mathematics, Computer Science. – 2016. –P. 26-35.

28 Ходашинский, В.А. Дель, А.Е. Анфилофьев. Выявление вредоносного сетевого трафика на основе ансамблей деревьев решений // Управление, вычислительная техника и информатика. Доклады ТУСУРа, 2014. – № 2 (32). – С. 55-62.

29 Бондур В.Г. Современные подходы к обработке больших потоков гиперспектралььной и многоспектральной аэрокосмической информации // Исследование Земли из космоса. – 2014. – №1. – С. 5-7.

30 Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms// Phys. Rev. E. – 2008. – P. 81-93.

31 Wang H., Shan H., Banerjee A. Bayesian cluster ensembles // Statistical Analysis and Data Mining. – 2011. – Vol. 4. – N 1. – P. 54-70.

32 Pestunov I.A., Berikov V. B., Kulikova E. A., Rylov S.A. Ensemble of clustering algorithms for large datasets // Optoelectronics, Instrumentation and Data Processing. – 2011. – Vol. 47. – N3. – P. 245-252.

33 Fern, X.Z., Brodley, C.E. Clustering ensembles for high dimensional data clustering // In Proc. International Conference on Machine Learning. – 2003. – P.186-193.

34 Мера растояния // Искуственный интеллект. http://www.aiportal.ru/articles/autoclassification/measure-distance.html. (алынған күні 20.10.2019)

35 Черикбаева Л.Ш., Байсылбаева Қ.Д. “Өзгермелі арақашықтық метрикасы негізіндегі алгоритмдер”//Вестник КазНИТУ, - 2018. №2, – С. 99 - 103.

36 Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms// Phys. Rev. E. –2008. –P. 45-52.

37 Wu M., Scholkopf B. Transductive Classification via Local Learning Regularization // Artificial Intelligence and Statistics. – 2007. – P. 628-635.

38 Berikov V.B. Weighted ensemble of algorithms for complex data clustering // Pattern Recognition Letters. – 2014. – Vol. 38. – P. 99-106.

39 Camps-Valls G., Marsheva T., Zhou D. Semi-supervised graph-based hyperspectral image classification // IEEE Transactions on Geoscience and Remote Sensing. – V.45(10). – 2007. – P. 3044-3054.

40 Zhou D., Bousquet O., Lal T., Weston J., Scholkopf B. Learning with local and global consistency // In Advances in Neural Information Processing Systems. – 2003. – P. 321-328.

41 Belkin M., Niyogi P., Sindhwani V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples // J. Mach. Learn. Res. – V.7. –2006. – P. 2399-2434.

42 Berikov V.B. Construction of an optimal collective decision in cluster analysis on the basis of an averaged co-association matrix and cluster validity indices // Pattern Recognition and Image Analysis. –2017. –V. 27(2). – P. 153-165.

43 Amirgaliyev, EN; Mukhamedgaliev, AF. An optimization model of classification algorithms // USSR Computational mathematics and mathematical physics. – 1985. –V.6. – P. 95-98.

44 Nussipbekov A.K., Amirgaliyev E., Hahn M. Kazakh traditional dance gesture recognition // Journal of Physics: Conference Series. – 2014. – Vol. 495. – Issue 1. – P. 32-40.

45 Dyusembaev A. E., Grishko M.V. Construction of a Correct Algorithm and Spatial Neural Network for Recognition Problems with Binary Data. Computational Mathematics and Mathematical Physics. – 2018. – Vol. 58. – P. 1673-1686.

46 Dyusembaev A.., Grishko M. On Correctness Conditions for Algebra of Recognition Algorithms with - Operators over Pattern Problems with Binary Data // DOKLADY MATHEMATICS. – V.98. – 2018 . – P. 421-424.

47 Kenshimov C., Bampis L., Amirgaliyev B., Arslanov M., Gasteratos A. Deep learning features exception for cross-season visual place recognition // Pattern recognition letters. – 2017. – V.100. –P.124-130.

48 Merembayev T., Yunussov R., Amirgaliyev Y. Machine Learning Algorithms for Stratigraphy Classification on Uranium Deposits // Procedia Computer Science. – 2019. – V.1. – P. 46-52.

49 Amirgaliev E., Isabaev Z., Iskakov S., Kuchin Y., Muhamediyev R., Muhamedyeva E., Yakunin K. Recognition of rocks at uranium deposits by using a few methods of machine learning // Advances in Intelligent Systems and Computing. – 2014. – V. 273. – P. 33-40.

50 Yu G. X., Feng L., Yao G. J., Wang, J. Semi-supervised classification using multiple clusterings // Pattern Recognition and Image Analysis. – 2016. –P. 681-689.

51 Berikov V., Karaev N., Tewari A. Semi-Supervised Classification with Cluster Ensemble // Proceedings of 2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). – Novosibirsk, Russia, 2017. – P. 245-250.

52 Berikov V. B., Amirgaliyev Y.N., Cherikbayeva L.Sh, Yedilkhan D., Tulegeniva B. “Classification at incomplete training information: usage of group clustering to improve performance” Journal of Theoretical and Applied Information Technology. – 2019. – Vol.97. – № 19. – P. 5048-5060.

53 Belkin M., Niyogi P., Sindhwani V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples // J. Mach. Learn. Res. – 2006. – Vol. 7. – P. 2399-2434.

54 Mercer J. Functions of positive and negative type and their connection with the theory of integral equations // Philosophical Transactions of the Royal Society A. – 1909. – P. 415–446.

55 Vishnoi, Nisheeth K. Lx=b Laplacian Solvers and Their Algorithmic Applications // Foundations and Trends in Theoretical Computer Science 8.1–2. – 2013. – P. 137-141.

56 Hyperspectral Remote Sensing Scenes. http://www.ehu.eus/ccwintco/index.php title= Hyperspectral_ Remote_Sensing_Scenes (дерек алынған күні: 17.04.2019).

57 Berikov V., Cherikbayeva L. Searching for Optimal Classifier Using a Combination of Cluster Ensemble and Kernel Method // Optimization Problems and Their Applications (OPTA-2018), CEUR Workshop Proceedings. – Omsk, Russia, 2018. – Vol. 2098. – P. 45-60.

58 Бериков В.Б., Амиргалиев Е.Н., Черикбаева Л.Ш. Полуконтролируемое обучение на основе кластерного ансамбля // Материалы II Международной научной конференции «Информатика и прикладная математика». – Алматы, Казахстан, 2017. – С. 65-76.

59 Черикбаева Л.Ш., Калдыбекұлы Б. Кластерлік талдауда топтық шешудің тиімді параметрлерін таңдау алгоритмдері // Материалы III Международной научной конференции «Информатика и прикладная математика». – Алматы, Казахстан, 2018. – С. 42-47.

60 Черикбаева Л.Ш. “Алгоритмдердің топтық шешімдерін пайдалана отырып тиімді классификаторларды іздеу” // Вестник КазНИТУ. - 2019. №2, – С. 289 - 292.

61 Neves Renata F.P. An Efficient Way Combining SVMs for Handwritten Digit Recognition / Renata F.P. Neles, Cleber Zanchettin, Alberto N.G. Lopes Filho // Artifical Neural Networks and Machine Learning (ICANN 2012). – 2012. – Vol. 7553. – P. 229-237.

62 Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, QiweiYe1, Tie-Yan Liu//LightGBM:Ahighly Efficient Gradient Boosting Decision Tree// 31st Conference on Neural Information Processing Systems (NIPS 2017). – Long Beach, CA, USA, 2017. – P. 58-67.

63 Tianqi Chen, Carlos Guestrin // XGBoost: A Scalable Tree Boosting System. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data MiningAugust. – 2016. – P. 785–794.

64 Sh. Shamiluulu, B. Y. Amirgaliyev, L. Cherikbayeva. Critical analysis of scikit-learn ml framework and weka ml toolbox over diabetes patients medical data // News of the National Academy of Sciences of the Republic of Kazakhstan, Series of Geology and Technical Sciences. – 2017. – №6. –P. 231-236.

65 Yelbig K., Treitel S. (2001) Computational Neural Networks For Geophysical Data Processing. Poulton M. M. (Eds.). –2001. –V.30. – P. 25-32.

66 Borsaru M., Zhou B., Aizawa T., Karashima H., Hashimoto T. (2006) Applied Radiation and Isotopes. – 2006. – V. 64(2). –P. 272–282.

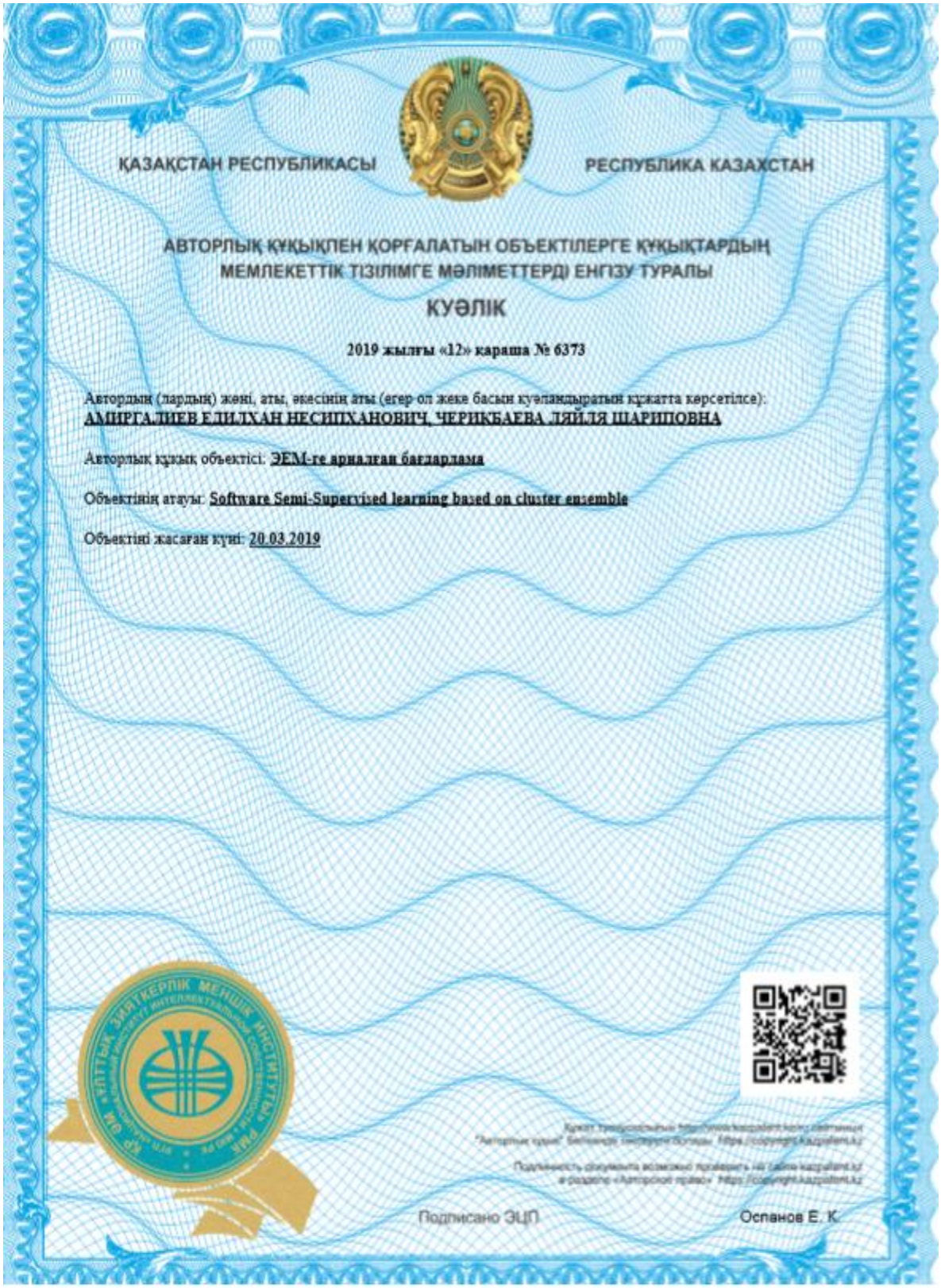
67 Викентьев А.А., Серов М.С., Бериков В.Б., Черикбаева Л.Ш., Тулегенова Б.А. “Коллективные расстояние для кластеризации множеств формул N-значной логики”. // Материалы IV Международной научно-практической конференции «Информатика и прикладная математика». – Алматы, Казахстан, 2019. – С. 219-234.

68 Шлезингер М.И. О самопроизвольном различении образов. – Киев: Наукова думка, 1965. – С. 62-70.

69 Айзерман М.А., Браверман Э.И., Розаноэр Л.И. Метод потенциальных функций в теории обучения машин. – М.: Наука, 1970. – 384с.

70 Цыпкин Я.З. Основы теории обучающихся систем. – М.: Наука, 1970. -252 с.

ҚОСЫМША А – Авторлық куәлік



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2019 жылғы «12» қараша № 6373

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
АМИРГАЛИЕВ ЕЛИЛХАН НЕСИПХАНОВИЧ, ЧЕРИКБАЕВА ЛЯЙЛЯ ШАРИПОВНА

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Software Semi-Supervised learning based on cluster ensemble

Объектіні жасаған күні: 20.03.2019

Подписано ЭЦП

Оспанов Е. К.

ҚОСЫМША Б – Топтық шешім алгоритмінің программа листингі

```
import numpy as np
from sklearn.base import ClassifierMixin, BaseEstimator
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import SpectralClustering
from sklearn.cluster import KMeans
from sklearn.neighbors import DistanceMetric
from sklearn import svm
from decimal import Decimal
from collections import Counter
import math
def my_kernel(X, Y):
    return 1-DistanceMetric.get_metric('euclidean').pairwise(X,Y)
class SSL(ClassifierMixin, BaseEstimator):
    def __init__(self, c):
        self.c = c
        self.cmatr = np.zeros((0,0))
    def cmatrix(self, X):
        n, m = X.shape
        res = np.zeros((n,2*self.c))
        centroids =np.zeros((n,2*self.c))
        for k in range(2,2+self.c):
                kmeans = KMeans(n_clusters=k)
                kmeans.fit(X)
        #agclus = AgglomerativeClustering(n_clusters=k,linkage = 'ward')
                #agclus.fit(X)
                specclus = SpectralClustering(n_clusters=k,
                eigen_solver='arpack',affinity="nearest_neighbors")
                specclus.fit(X)
                centroids = kmeans.cluster_centers_
                for i in range(n):
                        res[i,k-2] = kmeans.labels_[i]
                        res[i,k+self.c-2] = specclus.labels_[i]
                        #res[i,k+self.c-2] = agclus.labels_[i]
        self.cmatr = res
        return res,centroids
    def fitNN(self, Xl, labels):
        res = np.zeros(self.cmatr.shape[0],object)
    distances = my_kernel(self.cmatr,Xl)
        for i in range(len(res)):
                res[i] = labels[np.argmax(distances[i])]
        return res
    def fitSVM(self, Xl, labels):
```

```
            clf = svm.SVC(kernel=my_kernel, cache_size = 800)

            clf.fit(Xl, labels)
            return clf.predict(self.cmatr)
list_readed = []
with open('input.txt') as fin:
      for line in fin:
            params = line.split()
            temp_list = []
            for param in params:
                  prop = Decimal(str(param).strip())
                  temp_list.append(prop)
            list_readed.append(temp_list)
def ffind(t,w):
    if t in w:
       return 1
    else:       return 0;
# show to user readed values
X = []
for i in range(len(list_readed)):
      obj_str = ""
      xi =[]
      for list_el in list_readed[i]:
            obj_str += str(list_el) + "  "
            xi.append(float(list_el))
      X.append(xi)
      print("Object " + str(i+1) +": "+ obj_str)
ssl = SSL(5)
X = np.array(X)
X_train = split( random_state=42)
cmat, centr = ssl.cmatrix(X)
labes = np.array([[cmat[j,i] for j in range(len(cmat))]for i in range(7) ])
labes = np.array(labes)
labes = np.array(labes.T)   print(cmat.shape)
Xl = X[:,:10]    res = ssl.fitNN(Xl,labes)
print(res)     res1 =[ ]
for i in range(7):
      res = ssl.fitSVM(Xl,labes[:,i])
      res1.append(res)
print(res1)
```

ҚОСЫМША В – Ғылыми-зерттеу жұмысының нәтижесін енгізу актісі

БЕКІТЕМІН
Ақпараттық технологиялар және есептеуіш технологиялар институты бас директорының орынбасары, PhD, ассоц-проф. Мамырбаев О.Ж.
Мерзімі: 18.12. 2019

Ғылыми-зерттеу жұмыстарының нәтижесін енгізу актісі

1. Ғылыми- - зерттеу нәтижелерінің атауы: «Тану есептерінде топтық шешімдердің тиімді алгоритмдері мен ЭВМ бағдарламасы»

2. Енгізу нысанының авторы: Черикбаева Ляйла Шәріпқызы

3. Қысқаша аңдатпа: Классификациялау және тану есептерінде топтық шешімдер алгоритмдеріне зерттеу және талдау жүргізілді. Классификациялау және тану есептерінде жаңа топтық шешімдер алгоритмдерін зерттеліп құрылды. Топтық тану мәселелерінің қойылуында аясында жартылай бақылау арқылы оқыту және тану есебі зерттей отырылып шешілді. Базалық алгоритмдер тобында орталық объектілерді оқшаулауға негізделген топтық шешім табу алгоритмі алынды. Заманауи ақпараттық технологияларды қолдана отырып жаңа топтық шешімдер әдістерінің нәтижелері негізінде танудың ақпараттық жүйесін құрылды. Топтық шешімдер алгоритмдерінің нәтижелеріне талдау және бағалау жүргізілді.

4. Енгізу әсері: ғылыми жұмыстарды жүргізуде тиімді ғылыми негізделген әдістер мен алгоритмдер №AP05132648 «Заманауи сөйлеу және мобильді технологиялары негізінде вербальді-интерактивті роботтарды құру» гранттық жобасына машиналық көру және сөйлеулерді тану есептерінде бейне тану әдістері бойынша ғылыми әсер беретіндігі расталып жоба жұмысына енгізілді.

5. Енгізу орны мен уақыты: 050010. Алматы қаласы. ҚР БжҒМ, Ақпараттық және есептеуіш технологиялар институты. Робототехника және жасанды интеллект ғылыми зертханасы. Уақыты- 19 желтоқсан 2019 жыл.

6. Енгізу нысаны:

1) Топтық тану мәселелерінің қойылуында аясында жартылай бақылау арқылы оқыту және тану есебін шешу алгоритмі.

2) Базалық алгоритмдер тобында орталық объектілерді оқшаулауға негізделген топтық шешім табу алгоритмі.

3) Заманауи ақпараттық технологияларды қолдана отырып жаңа топтық шешімдер әдістерінің нәтижелері негізінде танудың ақпараттық жүйесі.

4) Авторлық құқықпен қорғалған объектілерге құқықтардың мемлекеттік тізіміне енгізу туралы куәлік. ЭВМ - ге арналған бағдарлама «Software Semi-Supervised leering on cluster ensemble» , берілу мерзімі 12 қараша 2019ж.

Енгізу комисиясы мүшелерінің қолдары:

АТжЕТ институты бас директорының орынбасары ф.-м.ғ.к., ассоц-проф. **Калижанова А.У.**
Зертхана меңгерушісі, т.ғ.д., проф. **Амиргалиев Е.Н.**
Аға ғылыми қызметкер, PHD **Қозбақова А.Х.**