

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ
ГРУПОВОГО ВРАХУВАННЯ АРГУМЕНТІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF THE
GROUP METHOD OF DATA HANDLING

Виконав: студент(ка) 2 курсу, групи КНТ-219м
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Біловоденко К.С.

(прізвище та ініціали)

Керівник Скрупський С.Ю.

(прізвище та ініціали)

Рецензент Зеленьова І.Я.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Біловоденка Кирила Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методу групового врахування аргументів. Research and Software Implementation of the Group Method of Data Handling

керівник проєкту (роботи) Скрупський Степан Юрійович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “03” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 5 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Вибір та обґрунтування структури системи, яка проєктується. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора. 6. Економіко-організаційна частина. 7. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Скрупський С.Ю., доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтролер	Калініна М.В., асистент		

7. Дата видачі завдання “04” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 5
7	Розробка економіко-організаційної частини.	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та безпеки в надзвичайних ситуаціях.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14-15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис) Біловоденко К.С.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Скрупський С.Ю.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
92 с., 19 рис., 6 табл., 4 додатки, 37 джерел.

МЕТОД ГРУПОВОГО ВРАХУВАННЯ АРГУМЕНТІВ,
РЕГРЕСІЙНИЙ АНАЛІЗ, МЕТОД РЕДУКЦІЇ ДАНИХ, КРИТЕРІЙ
ГРУПУВАННЯ ІНФОРМАТИВНИХ ОЗНАК, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес редукції даних.

Предмет дослідження – метод групового врахування аргументів.

Мета роботи – дослідження та розробка програмних засобів для реалізації методу групового врахування аргументів.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мова програмування Matlab, персональний комп'ютер з монітором, клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи.

Результати. Створено програмні засоби для редукції даних шляхом використання методу групового врахування аргументів. Розроблена програмна система може успішно функціонувати на комп'ютерах під управлінням операційних систем та використовуватися для відбору інформативних ознак.

Висновки. Розроблено програмне забезпечення для редукції даних на основі методу групового врахування аргументів. Проаналізовано предметну область, оформлено технічне завдання, обрано середовище розробки, створено структурну схему програмного забезпечення, виконано конструювання програмного забезпечення для реалізації методу групового врахування аргументів, здійснено тестування розробленої програми.

Галузь використання – редукція даних.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 92 pages, 19 figures, 6 tables, 4 appendixes, 37 sources.

METHOD OF GROUP ACCOUNTING OF ARGUMENTS, REGRESSION ANALYSIS, DATA REDUCTION METHOD, CRITERIA OF GROUPING OF INFORMATIVE FEATURES, SOFTWARE.

Object of study is a process of data reduction.

Subject of study is the method of group consideration of arguments.

The purpose of the work is to study and develop a software for the implementation of the method of group consideration of arguments.

Materials, methods, and tools: structured and object-oriented programming, programming language Matlab, keyboard, computer with a monitor, mouse manipulator.

Results. The software for data reduction has been developed. The developed software system can successfully operate on personal computers running operating system and can be used for informative feature selection.

Conclusions. Software for data reduction based on the method of group consideration of arguments has been developed. The subject area is analyzed, the technical task for program realization is developed, the development environment is chosen, the structural scheme of the program is developed, the software for realization of a method of group consideration of arguments is executed, the developed software is tested.

The field of use is data reduction.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Постановка задачі редукції даних	10
1.2 Аналіз методу групового врахування аргументів.....	12
1.3 Багаторядний алгоритм МГВА.....	14
1.5 Висновки за розділом 1	18
2 Обґрунтування вибору структури системи, що проєктується.....	20
2.1 Дослідження та аналіз вимог до програмної реалізації методу групового врахування аргументів.....	20
2.2 Вибір інструментарію розробки програмного забезпечення	21
2.3 Структурна схема програми.....	22
2.4 Висновки за розділом 2	23
3 Основні рішення щодо реалізації компонентів системи.....	25
3.1 Розробка модифікованого методу групового врахування аргументів ..	25
3.2 Функціональна схема програми	27
3.3 Тестування програмної реалізації методу групового врахування аргументів	29
3.4 Висновки за розділом 3	30
4 Керівництво програміста.....	31
4.1 Призначення та умови використання програмного забезпечення	31
4.2 Характеристики програмного забезпечення	31
4.3 Звернення до програми.....	32
4.4 Вхідні та вихідні дані програмного забезпечення	32
4.5 Повідомлення	32
5 Керівництво оператора	33
5.1 Призначення програмного забезпечення.....	33
5.2 Умови виконання програмного забезпечення.....	33

5.3 Виконання програмного забезпечення	34
5.4 Приклад роботи з програмою	34
5.5 Повідомлення оператору	36
6 Економіко-організаційна частина.....	37
6.1 Планування розробки програмного продукту.....	37
6.2 Визначення витрат на розробку програми	38
6.3 Розрахунок економічної ефективності програмного продукту.....	45
7 Охорона праці та безпека в надзвичайних ситуаціях.....	48
7.1 Аналіз потенційних небезпек.....	48
7.2 Заходи із забезпечення безпеки	49
7.3 Заходи з виробничої санітарії і гігієни праці	50
7.4 Заходи безпеки в надзвичайних ситуаціях	56
7.4.1 Заходи з пожежної безпеки	56
7.4.2 Заходи з цивільного захисту	57
Висновки	59
Перелік джерел посилання	61
Додаток А Технічне завдання	65
Додаток Б Опис програми	69
Додаток В Текст програми	73
Додаток Д Слайди презентації.....	85

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

МГВА – метод групового врахування аргументів;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

ПП – програмний продукт;

ШІ – штучний інтелект.

ВСТУП

В наш час задача моделювання природних та технологічних процесів набуває важливого значення в умовах необхідності економії часових та апаратних витрат на їх рішення. Це, в свою чергу, дало поштовх для детального вивчення механізму визначення та аналізу вхідної інформації, що її може бути застосовано для встановлення залежностей у процесах реальної системи.

Для зменшення розмірності задачі моделювання класичним прийомом є відбір інформативних ознак, який виділяє у просторі ознак лише ті, що мають суттєвий вплив на відгук. Саме їх використання забезпечує побудову адекватної моделі при мінімальній розмірності задачі моделювання [1]-[18].

Одним з методів відбору інформативних ознак є метод групового врахування аргументів. Метод групового врахування аргументів (МГВА) використовується в найрізноманітніших областях для аналізу даних і знаходження знань, моделювання систем, прогнозування, оптимізації й розпізнавання образів [6]-[9].

Розгляд цього методу та можливих його модифікацій представлений у даній роботі.

Для досягнення мети роботи необхідно розв'язати завдання:

- аналіз методу групового врахування аргументів;
- розробка технічного завдання;
- вибір середовища створення програмного забезпечення;
- розробка модифікованого методу групового врахування аргументів;
- конструювання програмного забезпечення для редукції даних шляхом використання методу групового врахування аргументів;
- тестування створеного програмного забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка задачі редукції даних

Процес редукції даних пов'язаний зі скороченням заданої навчальної вибірки та зменшенням кількості її екземплярів або ознак шляхом видалення неінформативних елементів чи групування деяких з них [1]-[18].

Нехай задана вибірка вихідних даних (1.1):

$$S = \langle X, Y \rangle, \quad (1.1)$$

де $X = \{X_1, X_2, \dots, X_L\} = \{X_i\}$ – вихідний набір значень ознак, що характеризують досліджуваний об'єкт або процес;

$Y = \{y_1, y_2, \dots, y_m\} = \{y_j\}$ – масив значень вихідного параметра в заданій вибірці;

$X_i = \{x_{ij}\}$ – i -та ознака у вихідній вибірці, $i = 1, 2, \dots, L$;

x_{ij} – значення i -ої ознаки для j -го екземпляра s_j вибірки, $j = 1, 2, \dots, m$;

y_j – значення прогнозованого параметра для j -го екземпляра;

L – загальна кількість ознак у вихідному наборі;

m – кількість екземплярів вибірки.

Тоді постановка задачі редукції даних може бути представлена одним з таких способів [1]-[18]:

– виділити комбінацію ознак X^* та екземплярів S^* з вихідного масиву даних, при якій досягається мінімум заданого критерію оцінювання набору ознак. Приклад наведено за формулою (1.2):

$$J(X^*, S^*) = \min_{X \in XS} J(Xe, Se), \quad (1.2)$$

де Xe та Se – елементи множини XS та SE ;

$J(Xe, Se)$ – критерій оцінювання значимості редукованого набору даних, що складається з набору ознак Xe та набору екземплярів Se ;

XS – множина всіх можливих комбінацій ознак, отримана з вихідного набору ознак X ;

SE – множина всіх можливих екземплярів, отримана з вихідної множини $S = \langle X, Y \rangle$.

– знайти мінімальний набір даних, що забезпечує досягнення заданого значення критерію оцінювання значимості набору даних за формулою (1.3):

$$\left| \langle X^*, S^* \rangle \right| = \min_{\substack{Xe \in XS, Se \in SE \\ J(Xe, Se) < \varepsilon}} \left| \langle Xe, Se \rangle \right|, \quad (1.3)$$

де ε – задане значення критерію J оцінювання інформативності набору даних.

Результатом виконання процедури відбору ознак є оптимальний набір даних (набір ознак X^* та набір екземплярів S^*), що має достатню інформативність J . Інформативність ознаки (набору ознак) – це величина, що відображає ступінь взаємозв'язку ознаки (набору ознак) із прогнозованим параметром. Інформативність комбінації ознак дорівнює сумі інформативності окремих ознак тільки при їхній незалежності [1]-[4].

Таким чином, отриманий у результаті редукції даних оптимальний набір $\langle X^*, S^* \rangle$, маючи достатню інформативність, найбільш повно відображає досліджуваний об'єкт або процес. При цьому з вхідного набору $S = \langle X, Y \rangle$ виключаються [1]-[4]:

- незначущі ознаки – ознаки, що не впливають на вихідний параметр;
- надлишкові ознаки – ознаки, значення яких залежать від інших ознак. Такі ознаки не приводять до поліпшення якості прогнозування по синтезованій моделі;
- незначущі та надлишкові екземпляри – екземпляри, що не впливають на якість побудованої розпізнавальної моделі.

Як правило, процес редукції даних полягає у відборі інформативних ознак та пошуку найбільш значущої комбінації ознак X^* . Після цього, за необхідністю, виконується виключення надлишкових екземплярів за допомогою тих же методів відбору ознак, транспонуючи при цьому дані вхідного набору $S = \langle X, Y \rangle$. Таким чином для розроблення ефективних методів редукції даних необхідно проаналізувати відомі методи відбору інформативних ознак [1]-[4].

Процес відбору інформативних ознак є досить важким і ресурсномістким завданням, оскільки він пов'язаний з необхідністю комбінаторного перебору. У наш час відомі різні методи виділення набору ознак [1]-[4]:

1.2 Аналіз методу групового врахування аргументів

Як відзначено вище, як базис для редукції даних доцільно обрати метод групового врахування аргументів [6]-[9].

Під методом групового врахування аргументів (МГВА) розуміють набір індуктивних алгоритмів МГВА, які дозволяють автоматично знаходити взаємозалежності у даних, а також вибирати оптимальну структуру мережі або моделі [6]-[9].

Цей підхід самоорганізації (вибору моделей оптимальної складності) відрізняється від дедуктивних методів. Такий підхід заснований на індуктивних принципах, застосування яких передбачає вирішення задач шляхом перебору з використанням зовнішнього критерію [6]-[9].

За допомогою перебору різних розв'язань методи індуктивного моделювання намагаються мінімізувати роль припущень автора у результатах моделювання. Комп'ютер сам знаходить структуру моделі і закони, що діють в об'єкті. Він може бути використаний при створенні штучного інтелекту як радник для вирішення суперечок та прийнятті рішень [6]-[9].

До методу групового врахування аргументів відносяться різні алгоритми, зокрема параметричні алгоритми, алгоритми кластеризації, ребінаризації та ймовірнісні алгоритми. Такий підхід базується на переборі моделей, які послідовно ускладнюються, а також на виборі найкращого рішення згідно мінімуму зовнішнього критерію. Як базисні моделі можуть використовуватися різні типи моделей, зокрема поліноміальні моделі, нелінійні, ймовірнісні функції, а також моделі кластеризації [6]-[9].

Алгоритми МГВА знайшли своє застосування для рішення різноманітних задач обробки експериментальних даних. Розроблено спектр параметричних (поліноміальних) та непараметричних (такі, що використовують кластеризації або аналоги) алгоритмів. Вибір алгоритму для практичного використання залежить від специфіки задачі, рівня дисперсії шуму, достатності вибірки, а також від того, чи містить вибірка виключно неперервні змінні [6]-[9]. Нижче наведено основні задачі, що вирішуються за МГВА [6]-[9]:

- ідентифікація фізичних закономірностей;
- апроксимація багатомірних процесів;
- короткостроковий покроковий прогноз процесів та подій;
- довгоостроковий покроковий прогноз;
- екстраполяція фізичних полів;
- кластеризація вибірки даних та визначення відповідної фізичної моделі об'єкта;
- розпізнавання образів у випадку неперервних або дискретних змінних;
- діагностика та розпізнавання за допомогою ймовірнісних переборних алгоритмів;
- самоорганізація багаторядних нейромереж з активними нейронами;
- нормативний векторний прогноз процесів;

– безмодельне прогнозування процесів за допомогою комплексування аналогів.

1.3 Багаторядний алгоритм МГВА

Припустимо, що існує набір вхідних даних у вигляді матриці X з n спостережень в просторі змінних, що варіюються, розмірністю $m > 1$, характерний для стандартної задачі множинної регресії. Нехай сформована навчальна послідовність прикладів, в якій у кожному рядку матриці X поставлено у відповідність відоме значення відгуку Y , виміряне у кількісній шкалі [6]-[9].

Необхідно, використовуючи методи самоорганізації, отримати модель, що виражає закон зміни відгуку Y в залежності від конкретних значень незалежних змінних X [6].

З однієї сторони, МГВА вважається інтелектуальним узагальненням регресійного аналізу, що розуміється у найширшому значенні. Від класичної множинної регресії МГВА відрізняється лише використанням специфічних квадратичних критеріїв зовнішнього або внутрішнього типу, а також багаторядними ітераційними процедурами знаходження оптимального рішення задачі.

З точки зору організації обчислень метод групового врахування аргументів можна представити наступним ітераційним циклом:

- задається деяка множина A достатньо простих функцій від вхідних аргументів, що називаються предикторами або частковими структурами моделі, і формується перший шар моделі;

- з часткових структур поточного шару генерується по визначеним правилам новий шар предикторів, які тепер самі стають останнім шаром;

- з часткових описів останнього шару відбираються L кращих, де L – ширина відбору (селекції);

- якщо не виконується умова зупину селекції (наприклад, продовжує зростати критерій якості моделі), здійснюється генерація нового шару;
- найкращий набір часткових структур останнього шару оголошується шуканим оптимальним рішенням задачі.

В цьому описі присутні усі ознаки еволюційного алгоритму – відбір (селекція) та генерація нового покоління [1]-[5].

Нарешті, форма багаторядного уявлення моделей МГВА, де у кожному шарі локалізуються достатньо прості функції (поліноми не більш як 2 порядку від двох змінних), але загальна цілісна модель представляє собою надзвичайно складну конструкцію, містить багато спільного з моделями штучних нейронних мереж, що описані нижче.

У літературі представлені різноманітні схеми процесу самоорганізації при синтезі моделей МГВА: комбінаторні, багаторядні, гібридизації, засновані на кінцевих стохастичних автоматах і т.п.

Зупинимося на загальній схемі багаторядного алгоритму МГВА, яка відтворює схему масової селекції, аналогічну задачі знаходження оптимальної структури персептрону. У багаторядній поліноміальній моделі «повний» опис (тобто регресійна модель від m факторів, формула (1.4))

$$y = F(x_1, x_2, \dots, x_m), \quad (1.4)$$

замінюється послідовністю рядів "часткових" описів:

- перший ряд селекції, наведений за формулою (1.5):

$$y_1 = f(x_1, x_2), y_2 = f(x_1, x_3), \dots, y_s = f(x_{m-1}, x_m), \quad (1.5)$$

де $s = C_m^2$;

- другий ряд селекції, наведений за формулою (1.6):

$$z_1 = f(y_1, y_2), z_2 = f(y_1, y_3), \dots, z_p = f(y_{s-1}, y_s), \quad (1.6)$$

де $p = C_s^2$ і т.д.

Загальна результуюча складність моделі залежить, таким чином, від двох факторів – виду часткового опису f і кількості рядів селекції.

Кожний частковий опис є функцією тільки двох змінних. Тому коефіцієнти такого регресійного рівняння можуть бути легко визначені навіть по невеликій кількості спостережень навчальної послідовності методом найменших квадратів. Різноманітні модифікації багаторядного алгоритму відрізняються один від одного видом опорної функції f . У алгоритмі з лінійними поліномами використовуються часткові описи виду, наведені на формулі (1.7):

$$Y_k = a_0 + a_1 \cdot x_i + a_2 \cdot x_j, \quad 0 < i < m, \quad 0 < j < m. \quad (1.7)$$

Ускладнення моделі в цьому випадку відбувається тільки за рахунок збільшення кількості аргументів, що враховуються: на першому ряді селекції синтезуються моделі, що містять по 2 аргументи, на другому – по 3 або 4, на третьому – до 8 аргументів і т.д.

Багаторядні алгоритми при використанні нелінійних опорних функцій, наприклад приведенні на формулі (1.8):

$$Y_k = a_0 + a_1 \cdot x_i + a_2 \cdot x_j + a_3 \cdot x_i \cdot x_j, \\ Y_k = a_0 + a_1 \cdot x_i + a_2 \cdot x_j + a_3 \cdot x_i \cdot x_j + a_4 \cdot x_i^2 + a_5 \cdot x_j^2. \quad (1.8)$$

дозволяють отримати моделі практично будь-якої складності, так як на кожному ряді селекції степінь поліному подвоюється. При цьому кількість коефіцієнтів моделі може обчислюватися вже мільйонами, хоча мінімум критерія селекції як правило досягається достатньо швидко.

З одного ряду селекції в інший на кожному кроці самоорганізації пропускаються не усі часткові описи, отримані шляхом повного перебору пар факторів (s, p і т.д.), а тільки невелика їх частина, наприклад, m рівнянь, які є «найкращими» з точки зору заданого критерію регулярності, що визначається по частковим описам. У якості конкретних математичних виразів, що використовуються для регуляризації, як правило використовується одна з наступних статистик:

– абсолютної середньоквадратичної помилки, наведеної на формулі (1.9):

$$\delta^2 = \frac{1}{n_{np}} \sum_{i=1}^{n_{np}} (y_i - y_i^*)^2; \quad (1.9)$$

– відносної середньоквадратичної помилки, наведеної на формулі (1.10):

$$\Delta^2 = \frac{n_{np} \delta^2}{\sum_{i=1}^{n_{np}} y_i^2}; \quad (1.10)$$

– коефіцієнта кореляції вихідної змінної Y з аргументом x_k , наведеного на формулі (1.11):

$$K_{yx_k} = \frac{\sum_{i=1}^{n_{np}} y_i x_{ik}}{\sqrt{\sum_i y_i^2 \sum_i x_{ik}^2}}, \quad (1.11)$$

де n_{np} – кількість точок вибірки;

Y і Y^* – фактичне і розрахункове значення прогнозуємої змінної.

Оскільки при використанні нелінійних опорних функцій помічається небезпека втрати суттєвого аргумента, то бажано використовувати алгоритми, оптимізуючі на кожному кроці довжину часткового опису.

Кількість рядів селекції рекомендується нарощувати до $s = (m - 1)$. На практиці ускладнення моделі зупиняють, коли подальше покращення критерію селекції не буде перевищувати деякого числа ε (параметр алгоритму) [9].

Варто відзначити, що в процесі оптимізації багаторядного алгоритму МГВА важливе місце займає критерій селекції наборів ознак, які корелюють між собою. Очевидно, що відбір наборів ознак на кожному ряді селекції, для яких значення помилки регресійного аналізу не перевищує жорстко заданого порогу, не забезпечує вибору дійсно суттєвих для групування ознак. Тому доцільною є розробки модифікації багаторядного алгоритму МГВА, в якому усувається зазначений недолік.

1.5 Висновки за розділом 1

Проаналізовано процес редукції даних. Надано постановку задачі редукції даних. Показано, що етап редукції даних є важливим при синтезі розпізнавальних та діагностичних моделей. Проаналізовано методи редукції даних. Визначено, що для редукції даних доцільно використовувати метод групового врахування аргументів.

Досліджено метод групового врахування аргументів. Відзначено, що в процесі оптимізації багаторядного алгоритму МГВА важливе місце займає критерій селекції наборів ознак, які корелюють між собою. Проте відбір наборів ознак на кожному ряді селекції, для яких значення помилки регресійного аналізу не перевищує жорстко заданого порогу, не забезпечує

вибору дійсно суттєвих для групування ознак. Тому доцільною є розробки модифікації багаторядного алгоритму МГВА, в якому усувається зазначений недолік.

2 ОБҐРУНТУВАННЯ ВИБОРУ СТРУКТУРИ СИСТЕМИ, ЩО ПРОЄКТУЄТЬСЯ

2.1 Дослідження та аналіз вимог до програмної реалізації методу групового врахування аргументів

Метою розробки є створення програмного забезпечення для реалізації методу групового врахування аргументів.

Діловий контекст розробки полягає в тому, що по завершенню реалізації проєкту власник матиме готовий програмний продукт, з якого зможе отримати матеріальну вигоду, автоматизувавши процес редукції даних.

Функціональними вимогами до програмного продукту (ПП) є такі:

- забезпечення файлового введення/виведення інформації;
- можливість додавання нових значень у вибірку;
- редукція даних за допомогою методу групового врахування аргументів;
- збереження і завантаження результатів редукції даних;
- наявність візуального інтерфейсу користувача.

Розроблений ПП має зберігатися на DVD диску розміром 8ГБ. Дані повинні зчитуватися на платформі Windows.

Інтерфейс має бути інтуїтивно зрозумілий користувачу. Колір меню не повинен дратувати око користувача.

Для запуску розроблюваного ПП треба:

- процесор частотою 2 ГГц;
- місце на жорсткому диску – 2 Гб;
- відеокарта з оперативною пам'яттю 64 Мб;
- оперативна пам'ять – 2 Гб;
- маніпулятор миша.

Розповсюдження програмного продукту повинно проводитись згідно з законам України.

2.2 Вибір інструментарію розробки програмного забезпечення

З метою розробки програмного забезпечення було проаналізовано різні технології та мови програмування [19]-[22]. Для створення програмного забезпечення, що реалізує метод групового врахування аргументів [6]-[9], обрано мову Matlab [21]-[22].

Середовище Matlab представляє собою пакет прикладних програм для рішення задач технічних обчислень, а також вбудовану мову програмування для здійснення технічних обчислень [21]-[22].

Порівняльний аналіз мови програмування Matlab та інших мов наведено у таблиці 2.1.

Таблиця 2.1 – Порівняльний аналіз мови програмування Matlab та інших мов програмування

Критерії порівняння	Мови програмування		
	Matlab	Python	C++
Наявність великої кількості бібліотек для математичних розрахунків	++	+	+
Наявність літератури та простота вивчення	++	++	++
Підтримка парадигми модульного програмування	++	+	-
Продуктивність роботи	++	++	+
Можливість паралельного програмування	++	+	+
Простота розробки інтерфейсу користувача	++	+-	+

2.3 Структурна схема програми

Для спрощення роботи системи та структуризації її окремих логічних та функціональних частин було організовано модульність.

Таким чином, система містить наступні модулі:

- Main.m, що містить первісну генерацію пар ознак, перерахунок порогового значення для відбору груп ознак та виведення кінцевих наборів ознак і спільних ознак у них;
- genData.m – відповідає за генерацію вхідних параметрів;
- regress.m – виконує регресійний аналіз для знаходження коефіцієнтів лінійної регресії в лінійних апроксимаціях функціональної залежності;
- repair.m – виконує формування нових груп ознак на кожній ітерації;
- getBestPairs.m – сортує групи ознак за величиною помилки апроксимації багатомірної моделі лінійною залежністю, сортує ознаки всередині кожного набору та вилучає повтори в них;
- gui.m – забезпечує графічний інтерфейс.

Виходячи з цього, було розроблено структурну схему, зображену на рис. 2.1.

Головний модуль Main.m поєднує роботу усіх підпорядкованих йому методів та звертається до них з метою делегування обов'язків та отримання результатів їх роботи.

В ньому виконується первісна генерація пар ознак, динамічний розрахунок критерію селекції для всіх ітерацій та визначення на його основі кількості нових груп ознак.

Обробка подій міститься в головному модулі.

В модулі genData.m виконується генерація вхідних ознак x та функції від них $-y$.

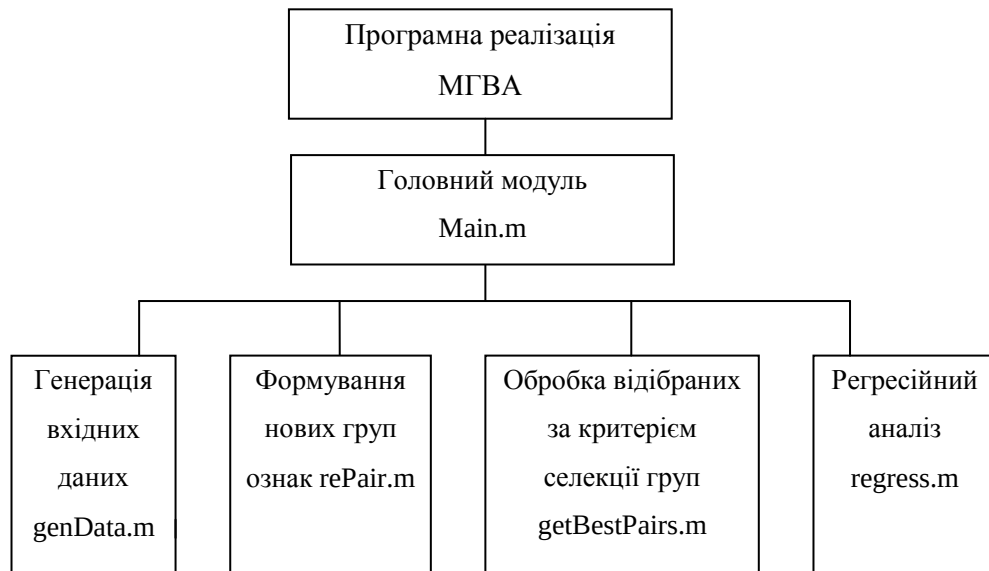


Рисунок 2.1 – Структурна схема програми

В модулі gePair.m відбувається побудова усіх можливих комбінацій нових груп ознак на основі отриманих на попередній ітерації.

В модулі обробки відібраних за критерієм селекції груп getBestPair.m реалізується сортування груп ознак після їх відбору згідно з пороговим значенням, а також сортування ознак всередині груп для виявлення та вилучення повторів з них.

В модулі регресійного аналізу regress.m виконується регресійний аналіз побудованих лінійних залежностей між групами ознак та відгуком для знаходження коефіцієнтів лінійної регресії та розрахунку корельованості ознак у рівнянні регресії.

В модулі графічного інтерфейсу gui.m реалізовано графічний інтерфейс системи, з яким безпосередньо спілкується користувач. В ньому ж забезпечені функції файлового введення/виведення даних, збереження вихідних даних в текстовому та графічному режимах.

2.4 Висновки за розділом 2

Досліджено та проаналізовано вимоги до програмного забезпечення для реалізації методу групового врахування аргументів.

Для розробки програмного забезпечення обрано мову пакету Matlab, яка є зручною для створення відповідної програми та містить набір вбудованих бібліотек, що спрощують реалізацію математичних обчислень.

Розроблено структурну схему програми. Визначено, що основними модулями розробленого програмного забезпечення є такі:

- Main.m, що містить первісну генерацію пар ознак, перерахунок порогового значення для відбору груп ознак та виведення кінцевих наборів ознак і спільних ознак у них;
- genData.m – відповідає за генерацію вхідних параметрів;
- regress.m – виконує регресійний аналіз для знаходження коефіцієнтів лінійної регресії в лінійних апроксимаціях функціональної залежності;
- repair.m – виконує формування нових груп ознак на кожній ітерації;
- getBestPairs.m – сортує групи ознак за величиною помилки апроксимації багатомірної моделі лінійною залежністю, сортує ознаки всередині кожного набору та вилучає повтори в них;
- gui.m – забезпечує графічний інтерфейс.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Розробка модифікованого методу групового врахування аргументів

Як відзначено вище, в процесі роботи багаторядного алгоритму МГВА важливе місце займає критерій селекції наборів ознак, які корелюють між собою. Проте відбір наборів ознак на кожному ряді селекції, для яких значення помилки регресійного аналізу не перевищує жорстко заданого порогу, не забезпечує вибору дійсно суттєвих для групування ознак.

Тому доцільною є розробки модифікації багаторядного алгоритму МГВА, в якому усувається зазначений недолік.

У запропонованій модифікації МГВА використовується модифікований критерій селекції, використання якого передбачає, що на кожному наступному селекційному ряді значення порогу зменшується за наступним виразом (3.1):

$$N_{i+1} = N_i / coef, \quad (3.1)$$

де N_i – значення порогу на поточному селекційному ряді;

$coef$ – коефіцієнт зменшення порогу.

Коефіцієнт $coef > 1$ задається користувачем. При цьому занадто великі значення цього коефіцієнту призведуть до швидкої зміни порогу до дуже малого числа, внаслідок чого методу не вдасться побудувати модель з прийнятною точністю. Дуже малі значення цього коефіцієнту будуть наближати реалізацію модифікованого МГВА до звичайного МГВА. Як приклад, для коефіцієнт зменшення порогу $coef$ можна встановлювати на рівні 1,61803399, що є числом золотого перетину.

Таким чином, у запропонованій модифікації МГВА на кожному наступному ряді селекції значення порогу поступово зменшується. Завдяки

цьому ми не розглядаємо слабо корельовані ознаки, що не можуть підлягати групуванню. Це скорочує кількість математичних обчислень та прискорює збіжність алгоритму до оптимальних груп інформативних ознак.

Крім того, було реалізовано рішення сортування наборів ознак на кожній ітерації алгоритму, що дозволяє вилучити повтори в наборах та легко інтерпретувати вихідні дані. Для зменшення впливу фактору випадковості в кінцевих групах ознак, виконується операція перетину груп, що вибирає спільні ознаки в наборах.

Виконаємо аналітичне порівняння базового та модифікованого МГВА. Для цього були обрані такі критерії: зміна критерію селекції, усунення некорельованих ознак з подальшого їх розгляду, швидкість збіжності, загальна кількість математичних обчислень, інформативність кінцевих груп ознак, інтерпретація результатів. Результати порівняння цих алгоритмів наведені в табл. 3.1.

Таблиця 3.1 – Порівняння багаторядних алгоритмів

Критерій/Алгоритм	Багаторядний алгоритм МГВА	Модифікований багаторядний алгоритм МГВА
1	2	3
Зміна критерію селекції	Залишається незмінним на кожній ітерації	Зменшується на кожній ітерації
Усунення некорельованих ознак з подальшого їх розгляду	Слабо контролюється	Передбачено шляхом зміни порогу відбору ознак у групи
Швидкість збіжності	Сумнівна	Забезпечується відкиданням повторів у групах ознак
Загальна кількість математичних обчислень	Велика за наявності повторюваних ознак у наборах	Скорочується за рахунок відкидання повторів у групах ознак
Інформативність кінцевих груп ознак	Невисока, оскільки включає невідкинуті некорельовані ознаки та повтори	Висока за рахунок операції перетину спільних ознак у групах

Продовження таблиці 3.1

1	2	3
Інтерпретація результатів	Дані складно сприймати, необхідна їх кінцева обробка	Зрозуміла за рахунок сортування та обробки кінцевих даних

Необхідно відзначити, що головним недоліком базового методу є спрощення розрахунку критерію селекції шляхом його жорсткого завдання впродовж всієї роботи методу. При цьому багато ознак, що можна відкинути в подальшій роботі алгоритму, залишаються «актуальними» для нього, а отже, зайвими у математичних обчисленнях.

Таким чином, запропоновано модифікацію МГВА, в якій, на відміну від існуючого методу, застосовується модифікований критерій селекції, використання якого передбачає, що на кожному наступному ряді селекції значення порогу поступово зменшується. Це дозволяє скоротити кількість математичних обчислень та прискорити збіжність алгоритму до оптимальних груп інформативних ознак.

3.2 Функціональна схема програми

Початок роботи з системою – це завантаження або генерація вхідних даних, а саме – навчальної вибірки (значення ознак та відгуку для екземплярів). Останні завантажуються з mat-файлу.

Після отримання вхідних параметрів, необхідно вказати кількість ітерацій алгоритму, або кількість рядів селекції.

На цьому етапі достатні умови для запуску роботи багаторядного алгоритму МГВА вже виконані. Необов'язковою є дія вказання виводу спільних ознак у кінцевих групах. Після запуску алгоритму, відбувається комбінація груп інформативних ознак, що можна об'єднати в одну ознаку без втрати точності побудови моделі на основі запропонованих ознак.

Функціональна схема розробленої програми представлена на рис. 3.1.

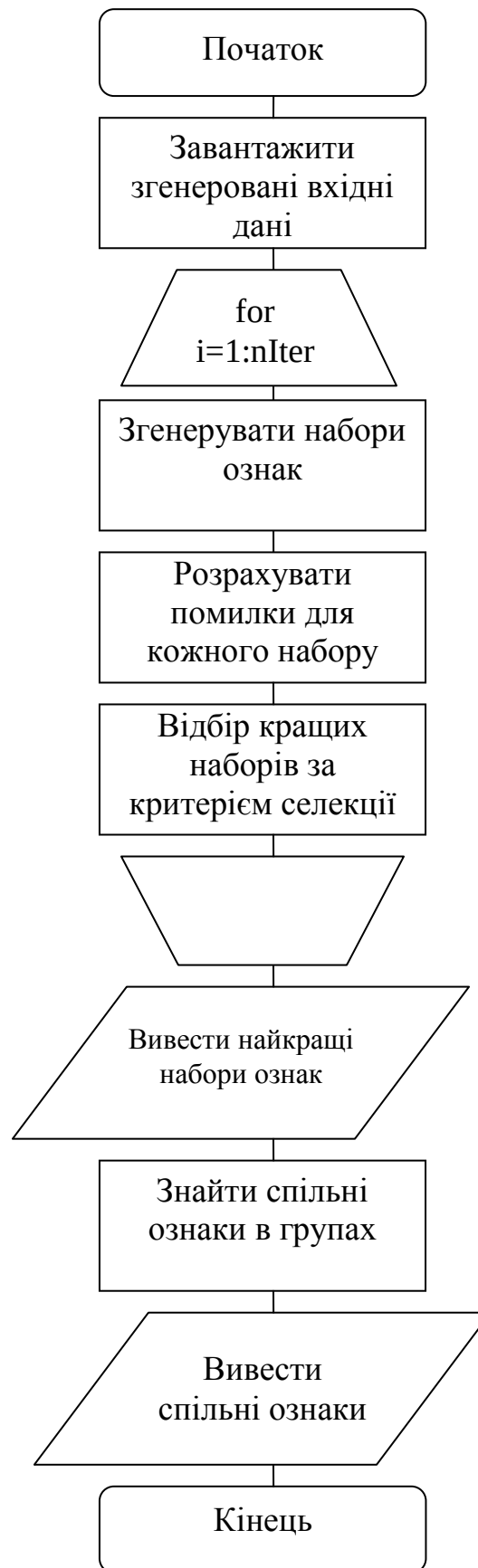


Рисунок 3.1 – Схема функціонування програми

Наприкінці роботи системи виконується дія графічного (у вікно програми) та текстового (створюється текстовий файл) виводу та збереження результатів роботи.

3.3 Тестування програмної реалізації методу групового врахування аргументів

В якості експериментального підтвердження правильності роботи програми було обрано дані метеорологічних досліджень (рис. 3.2).

Атмосферное давление, мм	758 756	757 755	756 754	755 753	754 752	753 751	752 750	752 750	753 751	754 752	755 753
Температура воздуха, °C	-1 -3	-2 -4	-4 -6	-1 -3	-1 -3	-4 -6	-6 -8	-4 -6	-3 -5	-5 -7	-6 -8
Влажность воздуха, %	45 40	55 50	60 55	54 49	45 40	47 42	55 50	77 72	65 60	75 70	86 81
Ветер, метр/сек	Ю-В 5-9	Ю-В 3-6	В 5-9	В 5-9	В 5-9	В 5-9	В 5-9	В 5-9	В 5-9	В 5-9	В 5-9
Комфорт, °C [?]	-5 -7	-6 -8	-7 -9	-7 -9	-5 -7	-9 -11	-10 -12	-11 -13	-7 -9	-9 -11	-9 -11

Рисунок 3.2 – Дані метеорологічних досліджень

З наведених даних можна бачити інформацію про температуру повітря, атмосферний тиск, швидкість вітру тощо. Відгуком на ці фактори є суб'єктивне сприйняття температури людиною (комфорт). Перед програмним продуктом постає завдання згрупувати близькі фактори з метою зменшення розмірності простору факторів.

Перш за все, метод повинен видавати передбачувані результати, оскільки обрана вибірка даних є інтуїтивно зрозумілою та очевидною (вибірку вважаємо репрезентативною). Крім того враховувався час проведення розрахунків та стабільність програми.

Програмний продукт в результаті обробки даних визначив наступні можливі угруповання: Температура вдень + Вологість повітря + Тиск вдень. У деяких групах до цих факторів ще додано температуру вночі або вологість

повітря вночі. З цих результатів можна зробити наступні висновки: суб'єктивне сприйняття погодних умов людиною є поєднанням впливу факторів, таких як температура, вологість повітря та тиск. При цьому для подальшої обробки інформації можна враховувати лише один з цих факторів або розглядати фактори як групу з метою зменшення розмірності задачі.

3.4 Висновки за розділом 3

Запропоновано основні рішення щодо реалізації програми. Запропоновано модифікацію МГВА, в якій, на відміну від існуючого методу, застосовується модифікований критерій селекції, використання якого передбачає, що на кожному наступному ряді селекції значення порогу поступово зменшується. Це дозволяє скоротити кількість математичних обчислень та прискорити збіжність алгоритму до оптимальних груп інформативних ознак.

Виконано тестування розробленої програмної реалізації методу групового врахування аргументів. Результати тестування підтвердили, що програма може використовуватися за призначенням.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення та умови використання програмного забезпечення

Програмне забезпечення призначений для реалізації методу групового врахування аргументів.

Для нормальної роботи програму необхідно запускати в операційній системі. Також повинен бути встановлений пакет Matlab.

Програмний продукт може бути використаний для формування груп інформативних ознак, що є корельовані між собою, з метою їх заміни однією ознакою і тим самим зменшення розмірності задачі моделювання залежності у від вектора x .

Для експлуатації програми потрібно мати таке програмно-технічне забезпечення:

- компютер зі встановленою операційною системою та встановленим пакетом Matlab;
- процесор частотою 2 ГГц;
- місце на жорсткому диску – 2 Гб;
- відеокарта з оперативною пам'яттю 64 Мб;
- оперативна пам'ять – 2 Гб;
- маніпулятор типу миша.

4.2 Характеристики програмного забезпечення

Програмне забезпечення розроблено за допомогою пакету Matlab з використанням графічного інтерфейсу використаного інструментарію. Для побудови системи використаний модульний підхід.

Робота з програмним забезпеченням може здійснюватися як в консольному режимі, так і у графічному режимі.

4.3 Звернення до програми

Для виконання програмного забезпечення потрібно відкрити з ОС файл Main.fig, в результаті чого буде запущено пакет Matlab і відкритий графічний інтерфейс. Другим способом є запуск програмного забезпечення з консолі за командою run ('Main'), проте для цього попередньо потрібно змінити поточний каталог на такий, в якому зберігається програма.

4.4 Вхідні та вихідні дані програмного забезпечення

Вхідними даними для програми є простір ознак та відповідний до них вектор y , що на практиці можна отримати в результаті збору даних після виконання серії експериментів.

Вихідними даними для програми є побудовані групи корельованих між собою ознак, що можна замінити однією ознакою для спрощення розмірності моделі функціональної залежності.

4.5 Повідомлення

Реалізована програмна система гарантує її стійке функціонування. У випадку виникнення помилок в програмі передбачено відображення стандартних діалогових вікон. Програма видає повідомлення у випадку введення даних некоректного формату, наприклад, замість числових даних, користувачем введено текстові.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програмного забезпечення

Програмне забезпечення призначено для реалізації методу групового врахування аргументів.

Для нормальної роботи програму необхідно запускати в операційній системі. Також повинен бути встановлений пакет Matlab.

Інтерфейс користувача програмного забезпечення забезпечує просту й ефективну взаємодію з користувачем.

Програмне забезпечення дозволяє виконувати користувачу такі функції:

- забезпечення файлового введення/виведення інформації;
- можливість додавання нових значень у вибірку;
- редукція даних за допомогою методу групового врахування аргументів;
- збереження і завантаження результатів редукції даних;
- наявність візуального інтерфейсу користувача.

5.2 Умови виконання програмного забезпечення

Для експлуатації програми потрібно мати таке програмно-технічне забезпечення:

- компютер зі встановленою операційною системою та встановленим пакетом Matlab;
- процесор частотою 2 ГГц;
- місце на жорсткому диску – 2 Гб;
- відеокарта з оперативною пам'яттю 64 Мб;
- оперативна пам'ять – 2 Гб;
- маніпулятор типу миша.

Час функціонування програмного забезпечення та отримання результату залежить від конфігурації комп'ютера.

5.3 Виконання програмного забезпечення

Для виконання програмного забезпечення потрібно відкрити з ОС файл Main.fig, в результаті чого буде запущено пакет Matlab і відкритий графічний інтерфейс. Другим способом є запуск програмного забезпечення з консолі за командою `run ('Main')`, проте для цього попередньо потрібно змінити поточний каталог на такий, в якому зберігається програма.

5.4 Приклад роботи з програмою

Після запуску програми з'явиться головне вікно (рис. 5.1).

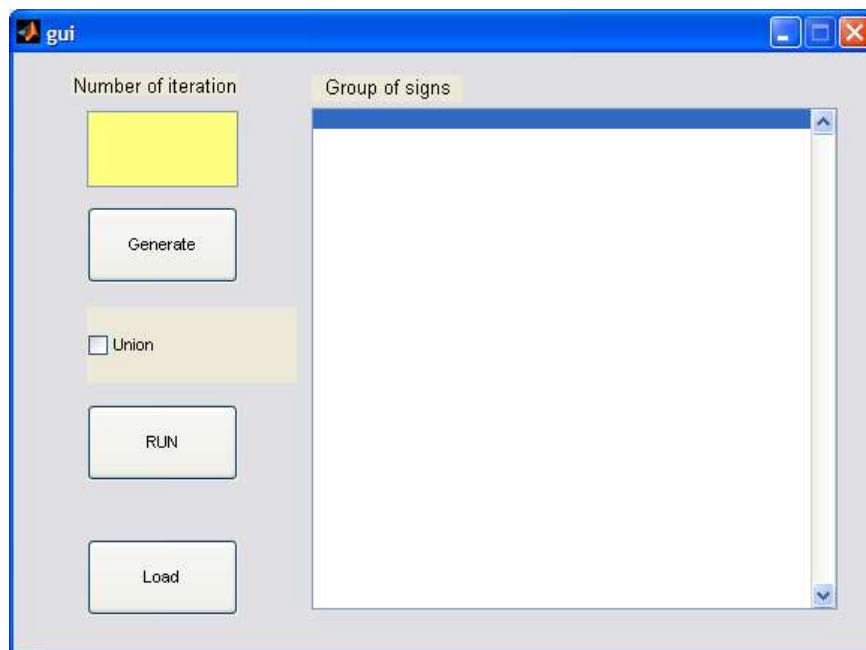


Рисунок 5.1 – Головне вікно програмного забезпечення

Після запуску програмного забезпечення необхідно згенерувати вхідні дані – ознаки x та вектор відгуків y . Для цього слід натиснути кнопку Generate і в файли `x.mat`, `y.mat` запишеться згенеровані вхідні вибірки x , y . У

випадку реальних даних вибірки будуть являти собою зібрану в результаті експериментів інформацію про ознаки та відгуки, і будуть зберігатися в текстовому форматі для їх використання у роботі програми.

Для коректної роботи методу необхідно вказати в текстовому полі Number of iterations кількість ітерацій методу, або кількість наборів селекційних рядів, що одне і те ж.

Для запуску роботи методу натиснути кнопку RUN. Після цього результати роботи методу – найоптимальніші групи ознак, запишуться у файл out.txt. Натискання на кнопку Load виведе вміст цього файлу у вікно форми. Після проведення таких операцій вікно форми буде мати вигляд, представлений на рис. 5.2.

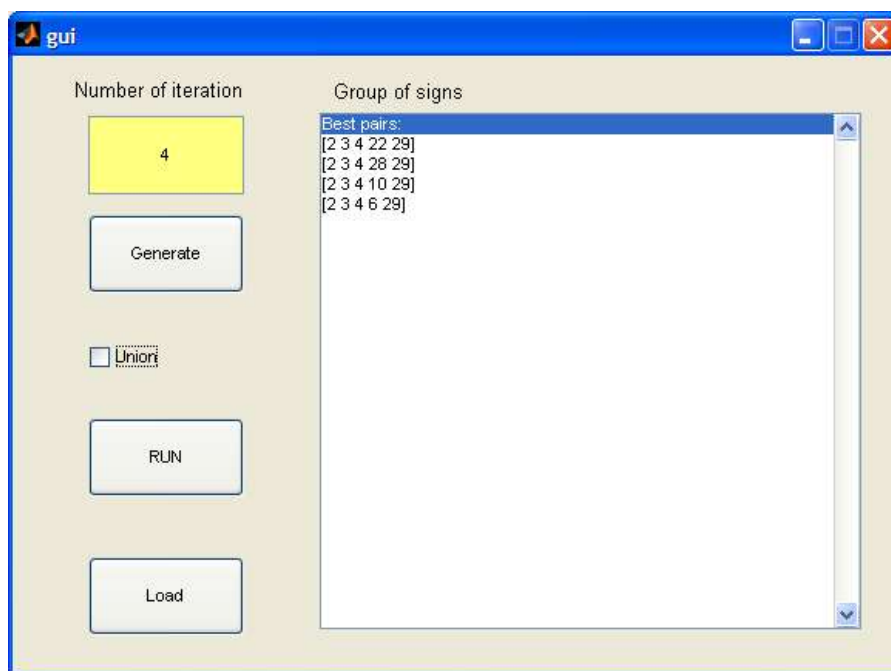


Рисунок 5.2 – Результати роботи програми

Завершити роботу з системою можна натисненням хрестика, що розташований справа вгорі вікна.

5.5 Повідомлення оператору

Реалізована програмна система гарантує її стійке функціонування. У випадку виникнення помилок в програмі передбачено відображення стандартних діалогових вікон. Програма видає повідомлення у випадку введення даних некоректного формату, наприклад, замість числових даних, користувачем введено текстові.

Крім того, до запуску роботи методу у користувача існує можливість встановити прапорець Union для відображення у вигляді повідомлення спільних ознак у вихідних групах ознак (рис. 5.3).

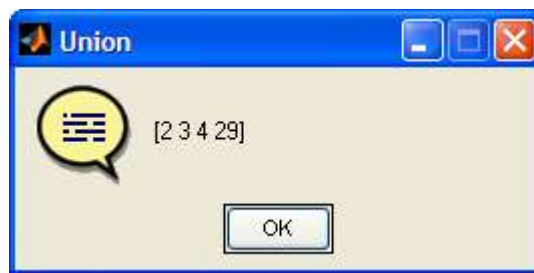


Рисунок 5.3 – Повідомлення про спільні ознаки у кінцевих групах ознак

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

У дипломній кваліфікаційній роботі магістра досліджено та реалізовано програму для редукції даних на основі методу групового врахування аргументів.

Очікується, що впровадження в експлуатацію розробленого програмного забезпечення на підприємстві дозволить скоротити одного співробітника, що відповідає за обробку великих масивів даних, з заробітною платою 7450 гривень на місяць.

У розробці беруть участь програміст із заробітною платою 15900 грн. в місяць і консультант з заробітною платою 7950 грн. у місяць. На розробку проєкту було витрачено 2 місяці. Розробка програми розпочалася першого вересня і повинна бути виконана до кінця жовтня 2020 року.

6.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на етапи [25]. Для кожного етапу вказуються трудомісткість, кількість виконавців і тривалість робіт. Інформація з характеристики робіт по розробці програмного забезпечення зведена в табл. 6.1.

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд-дн	%		
1	2	3	4	5
Аналіз ПЗ	8	11,9	Програміст Консультант	4 4
Визначення вимог до ПП	16	23,9	Програміст Консультант	8 8
Проектування структури програми	12	17,9	Програміст Консультант	6 6

Продовження таблиці 6.1

1	2	3	4	5
Розробка схеми функціонування програми	5	7,5	Програміст Консультант	3 2
Створення програмного коду	17	25,4	Програміст	17
Тестування і налагодження програмного продукту	5	7,5	Програміст	5
Складання програмної документації	4	6	Програміст Консультант	2 2
Разом	67	100	Програміст Консультант	45 22

6.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складають калькуляцію кошторисної вартості робіт, що включає статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);
- матеріали та комплектуючі вироби;
- витрати на спеціальне обладнання (на оплату машинного часу ЕОМ для написання і налагодження програмних засобів);
- накладні витрати.

Витрати за статтею «Основна заробітна плата» складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми.

Розрахунок зарплати на підставі даних про трудомісткість, представлений в табл. 6.2.

Таблиця 6.2 – Основна заробітна плата

Посада виконавця	Чисельність, чол.	місячний оклад	Кількість місяців роботи	Сума ЗП, грн.
Програміст	1	15900	2	31800
Консультант	1	7950	1	7950
Разом	2			39750

Додаткову заробітну плату приймають рівною 10% від основної зарплати. Вона розраховується за формулою (6.1):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 10\%. \quad (6.1)$$

Підставивши величину основної заробітної плати в дану формулу отримаємо: $ЗП_{\text{дод}} = 39750 \cdot 0,1 = 3975$ грн.

Відрахування на єдиний соціальний внесок (ЄСВ) визначають у відсотковому відношенні від сум основної та додаткової зарплати з урахуванням премій і доплат. Відрахування вираховуються за формулою (6.2):

$$ЄСВ = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22. \quad (6.2)$$

Розрахуємо:

$$ЄСВ = (39750 + 3975) \cdot 0,22 = 9620 \text{ грн.}$$

Використовується три найменування матеріалів: папір – 250 грн., картридж – 650 грн., флеш-носій – 220 грн. Витрати на матеріали розраховують за формулою (6.3):

$$M = \sum_{i=1}^n (\Pi_i \cdot N_i \cdot (1 + K_{\text{т.з.}}) - \Pi_{io} \cdot N_{io}), \quad (6.3)$$

де M – витрати на матеріали, закупні напівфабрикати і комплектуючі вироби, грн.;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Π_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн.;

N_i – потреба в i -му матеріалі, напівфабрикаті і комплектуючому;

Π_{io} – ціна зворотних відходів i -го найменування матеріалу, грн.;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

Значення Π_{io} , N_{io} , $K_{\text{т.з.}}$ наступні:

$$\Pi_{io} = 0; N_{io} = 0; K_{\text{т.з.}} = 0,05.$$

Розрахуємо:

$$M = (1 + 0,05) \cdot (250 + 650 + 220) = 1176 \text{ грн.}$$

Таким чином, разом витрати на матеріали та комплектуючі вироби становлять 1176 гривень.

У статтю «Витрати на спеціальне обладнання» входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання. Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів. Для цього необхідно скласти кошторис «витрат на утримання і експлуатацію устаткування», виходячи з якого визначиться вартість одного машино-години роботи ПК, після множення якої на машинний час пішло на

написання і налагодження програми отримаємо витрати на оплату машинного часу.

Амортизаційні відрахування визначають за формулою (6.4):

$$A = БВ \cdot \frac{H_a}{100}, \quad (6.4)$$

де БВ – балансова вартість обчислювальної техніки, грн.;

H_a – норма амортизаційних відрахувань на повне оновлення обчислювальної техніки, %. Приймаємо, що техніка повинна працювати 4 роки, тому $H_a = 25\%$.

Балансова вартість обчислювальної техніки становить 27080 грн. Тому амортизаційні відрахування становлять:

$$A = 27080 \cdot 0,25 = 6770 \text{ грн.}$$

Статтю "Експлуатація обладнання" розраховують підсумовуванням витрат на силову електроенергію і допоміжні матеріали. Витрати на електроенергію визначають за формулою (6.5):

$$B_e = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зч}} \cdot K_{\text{зп}} \cdot C_e, \quad (6.5)$$

де N_n – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{еф}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{\text{зч}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зп}}$ – коефіцієнт завантаження по потужності;

C_e – ціна одного кВт · год. електроенергії, грн. / (кВт · год.).

Номінальна Потужність ЕОМ – 0,2 кВт. Річний ефективний фонд часу роботи ЕОМ ставити 1800 годин. Середні коефіцієнти завантаження за часом

та за потужністю рівні відповідно 0,9 та 0,6. Ціна однієї кіловат-години електроенергії ставить 2,11 грн. Отримуємо:

$$B_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 410,18 \text{ грн.}$$

Приймаємо, що $B_e = 410$ грн.

Витрати за статтею "Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди" визначають за формулою (6.6):

$$ЗП_{\text{обсл}} = \Phi ЗП_r \cdot (1 + K_{\text{відрах}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}}, \quad (6.6)$$

де $\Phi ЗП_r$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн.;

$K_{\text{відрах}}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідне технічне обслуговування ЕОМ, годин / рік. Приймаємо $t_{\text{обсл}} = 36$ год.;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, годин / рік. Приймаємо, що річний ефективний фонд часу обслуговуючого персоналу становить 1800 годин.

Заробітна плата обслуговуючого персоналу = 8850 грн./місяць.

Розрахуємо:

$$\Phi ЗП_r = 8850 \cdot 12 = 106200 \text{ грн.}$$

$$ЗП_{\text{обсл}} = 106200 \cdot (1 + 0,22) \cdot 36 / 1800 = 2591 \text{ грн.}$$

Суму витрат за статтею "Поточний ремонт обладнання" обчислимо як 3% від балансової вартості обладнання, оскільки неможливо заздалегідь визначити, скільки коштів необхідно витратити на придбання запасних частин наявного обладнання. Балансова вартість обчислювальної техніки становить 27080 грн. Тому сума витрат за статтею "Поточний ремонт обладнання" становить: $27080 \cdot 0,03 = 812$ грн.

Інших витрат за проектом не передбачено, тому сума витрат за цією статтею дорівнює нуль гривень.

Кошторис витрат на утримання і експлуатацію устаткування наведений в табл. 6.3.

Таблиця 6.3 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація обладнання	6770
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	2591
Експлуатація обладнання (крім витрат на поточний ремонт)	410
Поточний ремонт обладнання	812
Інші витрати	0
Разом	10583

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за формулою (6.7):

$$B_{\text{МО}} = B_{\text{екс}} \cdot t_{\text{МО}}, \quad (6.7)$$

де $B_{\text{МО}}$ – витрати на оплату машинного часу, грн.;

$V_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино–год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино–год.

Розрахуємо експлуатаційні витрати:

$$V_{\text{екс}} = 10583/1800 = 5,88 \text{ грн. / машино–год.}$$

Для написання і налагодження даного програмного продукту ЕОМ експлуатується протягом 45 днів в одну зміну по 8 годин, що складає в сумі 360 годин. Таким чином отримуємо суму за статтею «Витрати на спеціальне обладнання»:

$$V_{\text{мо}} = 5,88 \cdot 360 = 2117 \text{ грн.}$$

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників. Для обчислення цієї статті видатків у дипломному проєкті приймемо, що накладні видатки становлять 24% від основної заробітної плати розробників, що складає 39750 грн. Тому сума накладних видатків становить: $39750 \cdot 0,24 = 9540$ грн.

Калькуляція кошторисної вартості робіт з розробки програми наведена в табл. 6.4.

Таблиця 6.4 – Калькуляція кошторисної вартості робіт з розробки програми

Найменування статей витрат	Сума, грн.
Основна заробітна плата	39750
Додаткова заробітна плата	3975
ЄСВ	9620
Матеріали і комплектуючі	1176
Витрати на спеціальне обладнання	2117
Накладні витрати	9540
Разом	66178

6.3 Розрахунок економічної ефективності програмного продукту

Річну економію від впровадження програми, що дозволяє знизити витрати машинного часу ЕОМ для рішення певного завдання в порівнянні із програмою, що застосовувалася раніше, визначають за формулою (6.8):

$$E = (T_{M_1} - T_{M_2}) \cdot B_{\text{екс}} + E_{\text{к}}^a \cdot \frac{\Phi_{\text{б.КТЗ}} \cdot (T_{M_1} - T_{M_2})}{\Phi_{\text{еф.КТЗ}}} - \frac{S_{\text{рп}}}{T_{\text{с}}} + E_{\text{зп}}, \quad (6.8)$$

де T_{M_1} , T_{M_2} – машинний час, необхідний для розв’язання поставлених завдань, відповідно в старому та у новому варіантах, машино–год./рік;

$B_{\text{екс}}$ – експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн/ машино–год.;

$E_{\text{к}}^a$ – нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{\text{б.КТЗ}}$ – балансова вартість ЕОМ (КТЗ), грн.;

$\Phi_{\text{еф.КТЗ}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год.;

$S_{\text{рп}}$ – сумарні витрати на розробку програми, грн.;

T_c – термін служби впроваджуваної програми до її морального зношування, років. Приймаємо цей термін 5 років;

$E_{\text{зп}}$ – економія на заробітній платі.

До впровадження даного ПЗ дані функції виконувались співробітником, що відповідає за обробку рукописних символів, 1800 годин на рік, а після 0 годин на рік (економія складає 1 ставка одного співробітника, що відповідає за обробку великих масивів даних, з заробітною платою 7450 гривень на місяць).

Розрахуємо річну економію за формулою (6.8):

$$E = (1800 - 0) \cdot 5,88 + 0,2 \cdot (27080 \cdot (1800 - 0)) / 1800 - 66178/5 + 7450 \cdot 1,0(1 + 0,22) \cdot 12 = 111832 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень $E_{\text{ф}}$ и строк їхньої окупності $T_{\text{ок}}$ розраховуються за формулами (6.9) і (6.10):

$$E_{\text{ф}} = \frac{E}{S_{\text{рп}}}, \quad (6.9)$$

$$T_{\text{ок}} = \frac{S_{\text{рп}}}{E}. \quad (6.10)$$

Розрахуємо:

$$E_{\text{ф}} = 111832 / 66178 = 1,7,$$

$$T_{\text{ок}} = 66178 / 111832 = 0,59 \text{ року.}$$

Зведемо отримані техніко–економічні показники в табл. 6.5.

Таблиця 6.5 – Техніко–економічні показники

Показники	Сума	Одиниця виміру
Витрати на розробку програми	66178	грн.
Річна економія	111832	грн.
Строк окупності програми	0,59	рік

Таким чином, впровадження в експлуатацію розробленого програмного забезпечення на виробничому підприємстві дозволить скоротити одного співробітника, що відповідає за обробку великих масивів даних, з заробітною платою 7450 гривень на місяць. Економічні розрахунки показали, що строк окупності програми 0,59 року, що значно менше часу її нормативного строку окупності (5 років). Крім того, річна економія при використанні програми становить 111832 грн. Виходячи з виконаних розрахунків, можна зробити висновок, що розробка даного програмного продукту є економічно доцільною.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

У ході виконання магістерської роботи досліджуються та програмно реалізуються методи групового врахування аргументів.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники:

- підвищені статичні навантаження користувачів ПК, які пов'язані з тривалим перебуванням в однотипній позі та повторних однотипних рухів кистями рук. Порушення правил ергономіки може призвести до ураження рухового апарату та тунельного синдрому;

- небезпеки, пов'язані електростатичним випромінюванням, що може призвести до загальних захворювань;

- можливість враження електричним струмом внаслідок порушень правил електробезпеки або несправності електроприладів, що може призвести до електричних травм або летального випадку;

- незадовільні параметри мікроклімату в приміщенні внаслідок не ефективної роботи апаратури для опалювання та повітрообміну, що може призвести до загальних захворювань;

- перевищення рівнів вібрації застарілого обладнання, що може призвести до сильної втоми, незначних змін функцій організму (порушення серцевої діяльності, нервової системи);

- недостатнє освітлення робочої зони внаслідок несправності освітлювальних приладів, що може призвести до погіршення зору;

- можливість загорянь внаслідок порушень правил пожежної безпеки, що може призвести до пожежі;

- неправильні дії персоналу під час надзвичайних ситуацій внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків.

7.2 Заходи із забезпечення безпеки

Для забезпечення ергономічності робочого місця згідно Нормативно-правового акту з охорони праці 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [26], де зазначена заборона захащеності робочого місця, дотримання нормативів щодо площини робочого місця на одного працівника, також вимоги до організації робочого місця зазначені в ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги» [27].

Було прийнято наступні рішення:

- оснащення офісних приміщень повинно бути виконано раціонально з точки зору ергономіки, зручне та раціональне розташування робочих місць визначається відстань між ними, згідно з вимогами, щодо виробничих приміщень площа одного робочого місця $\geq 6 \text{ м}^2$, а об'єм $\geq 20 \text{ м}^3$;

- конструкція робочого столу відповідає сучасним вимогам ергономіки і забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів;

- висота робочої поверхні робочого столу з візуальним дисплейним терміналом регулюються в межах 680...800 мм, а ширина і глибина - забезпечує можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина – 800...1000 мм);

- робочий стіл має простір для ніг заввишки не менше ніж 700 мм, завширшки не менше ніж 600 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги – ніж 650 мм;

- робочий стілець підйомно-поворотний, регульований за висотою, з кутом нахилу сидіння та спинки, висота спинки стільця становить 300 ± 20 мм, ширина – не менше ніж 380 мм, радіус кривизни горизонтальної площини – 400 мм, кут нахилу спинки має регулюватися в межах $1...30^\circ$ від вертикального положення. Відстань від спинки до переднього краю сидіння

має регулюватися в межах 260...400 мм [27].

За ступенем небезпеки ураження електричним струмом, відповідно до Правил улаштування електроустановок (ПУЕ), приміщення обчислювального центру (ОЦ) відноситься до категорії приміщень без підвищеної небезпеки [28].

Заходи по забезпеченню електробезпеки розроблено у відповідності до ПУЕ та ДСТУ 7237:2011 «Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту», за способом захисту людини від ураження електричним струмом – відеотермінали, ПК, периферійні пристрої ПК та устаткування для обслуговування, ремонту та налагодження ПК відповідають першому класу захисту та заземлені [29].

Для забезпечення електробезпеки у ОЦ використано наступні технічні заходи та засоби електробезпеки:

- ізоляція струмопровідних частин;
- недоступність струмопровідних частин;
- захисне відключення;
- ізоляція струмопровідних частин;
- захисне заземлення;
- блокування, занулення;
- захисне розділення електричних мереж;
- компенсація ємнісних струмів замикання на землю;
- вирівнювання потенціалів тощо [29].

7.3 Заходи з виробничої санітарії і гігієни праці

У обчислювальному центрі, згідно Державних будівельних норм В.2.5-28:2018 «Природне і штучне освітлення» [31] передбачене природне та штучне освітлення, штучне освітлення в приміщенні, здійснено системою загального рівномірного освітлення, значення освітленості на поверхні

робочого столу в зоні розміщення документів становить 300 лк, як джерела штучного освітлення в приміщенні застосовані люмінесцентні лампи типу ЛБ, при застосуванні яких дотримались наступних умов:

- температура навколишнього повітря не повинна бути нижче, ніж 5 °С;
- напруга на освітлювальних приладах повинна бути не менше, ніж 90 % номінальної [32].

Розрахуємо систему загального рівномірного штучного освітлення в офісному приміщенні.

Приймаємо розміри приміщення довжина $A = 20$ м., ширина $B = 13$ м., висота $H = 4,5$ м. Висота робочої поверхні складає $h = 0,7$ м. Коефіцієнти відбиття поверхонь приміщення: $\rho_c = 70\%$; $\rho_{ст} = 50\%$; $\rho_n = 30\%$.

Згідно з Державних будівельних норм В.2.5-28:2018 «Природне і штучне освітлення» рівень нормованого загального освітлення в приміщенні нормуються як $E_n = 300$ лк.

Обираємо світильник типу ЛВП, призначений для приміщень висотою до 4,5 м. $IP=54$ – ступінь захисту світильника від пилу та брызок води, $L/h = 1,3$ – числове значення коефіцієнта світильника.

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,4$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні за формулою (7.1):

$$N_p = \frac{B}{(H-h_p) \cdot [L/h]}, \quad (7.1)$$

$$N_p = \frac{B}{(H-h_p) \cdot [L/h]} = \frac{13}{(4,5-0,7) \cdot 1,3} = 2,6 \approx 3 \text{ ряди світильників.}$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (7.1):

$$L_{\max} = \frac{B}{N_p}, \quad (7.2)$$

де B – ширина приміщення, м;

N_p – кількість рядів.

Розрахуємо:

$$L_{\max} = \frac{13}{3} = 4,3 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (7.3):

$$h = \frac{L_{\max}}{[L/h]}, \quad (7.3)$$

$$h = \frac{4,3}{1,3} = 3,3 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі на основі формули (7.4):

$$h_3 = H - h_p - h, \quad (7.4)$$

де H – висота приміщення, м;

h_p – висота робочої поверхні, м;

h – висота підвісу світильника над робочою поверхнею, м.

Розрахуємо:

$$h_3 = 4,5 - 0,7 - 3,3 = 0,5 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (7.5):

$$i = \frac{A \cdot B}{h \cdot (A + B)}, \quad (7.5)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м.

Розрахуємо:

$$i = \frac{20 \cdot 13}{3,3 \cdot (20 + 13)} = 2,34.$$

Значення коефіцієнта використання світлового потоку η вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення (ρ_c , $\rho_{ст}$, $\rho_{п}$) та індексу приміщення. В нашому випадку $\eta = 58 \%$.

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні на основі формули (7.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_3 \cdot Z}{\eta}, \quad (7.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні,

лм;

E_n – нормоване значення освітленості, лк;

S – площа освітлюваної поверхні, m^2 ;

k_z – коефіцієнт запасу;

z – коефіцієнт нерівномірності (мінімальної) освітленості;

η – коефіцієнт використання світлового потоку.

Розрахуємо:

$$\Phi_{\Sigma} = \frac{300 \cdot 20 \cdot 13 \cdot 1,4 \cdot 1,1}{0,58} = 207104 \text{ лм.}$$

Визначаємо умовну загальну кількість світильників у приміщенні на основі формули (7.7):

$$N^* = \frac{A \cdot B}{L_{\max}^2}, \quad (7.7)$$

$$N^* = \frac{20 \cdot 13}{4,3 \cdot 4,3} = \approx 15 \text{ шт.}$$

Кратне числу рядів $N_p = 3$.

Розрахуємо світловий потік умовного джерела світла на основі формули (7.8):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм,} \quad (7.8)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні.

Розрахуємо на основі формули (7.9):

$$N_{\text{л}} = N^* \cdot n, \quad (7.9)$$

де n – кількість ламп у світильнику (в ЛВП світильнику знаходиться 2 лампи).

Розрахуємо загальну кількість ламп у приміщенні та розрахункове значення сумарного світлового потоку:

$$N_{\text{л}} = 15 \cdot 2 = 30 \text{ шт.},$$

$$\Phi_{\text{л}} = \frac{207104}{30} = 6904 \text{ лм.}$$

Обираємо тип стандартної лампи з найближчим значенням фактичного світлового потоку лампи $\Phi_{\text{л}}$ і знайдемо коефіцієнт m (співвідношення між розрахунковим світловим потоком лампи $\Phi_{\text{л}}^*$ та фактичним світловим потоком вибраної стандартної лампи $\Phi_{\text{л}}$):

Обираємо лампу ЛБ з $P_{\text{л}}=80$ Вт, $\Phi_{\text{л}} = 5400$ лм, використовуємо формулу (7.10):

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}}, \quad (7.10)$$

$$m = \frac{6904}{5400} = 1,28.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні на основі формули (7.11):

$$N = N^* \cdot m, \quad (7.11)$$

$$N = 15 \cdot 1,28 \approx 18 \text{ шт.}$$

Фактична кількість ламп у приміщенні обчислюється за формулою (7.12):

$$N_{\text{фл}} = N_{\text{ф}} \cdot n, \quad (7.12)$$

$$N_{\text{фл}} = 18 \cdot 2 = 36 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп за формулою (7.13):

$$E_p = \frac{\Phi_{\text{л}} \cdot N_{\text{фл}} \cdot \eta}{S \cdot k_3 \cdot z} \text{ лк}, \quad (7.13)$$

$$E_p = \frac{5400 \cdot 36 \cdot 0,58}{20 \cdot 13 \cdot 1,4 \cdot 1,1} = 282 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_{\text{н}}, \text{ лк.}$

7.4 Заходи безпеки в надзвичайних ситуаціях

7.4.1 Заходи з пожежної безпеки

Великою небезпекою для обчислювального центру є пожежа, відносно якої можна виділити наступне:

– обчислювальний центр наповнений різноманітними електричними приладами, а пожежа основний фактор для їх знищення, що може призвести до великих матеріальних втрат, так як в приміщенні обчислювального центру присутні всі три основні чинники, необхідні для виникнення пожежі: горючі речовини, окислення і джерела запалювання, то налагодження заходів

пожежної безпека в такому приміщенні є першочергово важливим;

- ймовірними причинами виникнення пожеги можуть бути несправність електрообладнання (кабелів, розеток), короткі замикання внаслідок виходу з ладу чи експлуатації несправного електроустаткування (персональних електронно-обчислювальних машин, периферійних пристроїв), порушення правил протипожежної безпеки тощо;

- для забезпечення пожежної безпеки проводився пожежна профілактика, яка включає в себе комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожежі, обмеження її поширення, а також на створення умов для успішного гасіння пожежі;

- експлуатація ліній електромережі практично повністю унеможлиблює виникнення електричного джерела загоряння в наслідок короткого замикання та перевантаження проводів, так як застосовуються дроти з важкогорючою і негорючою ізоляцією;

- для відводу теплоти від ПК у обчислювальному центрі постійно діє потужна система кондиціонування [35].

7.4.2 Заходи з цивільного захисту

Порядок дій сил цивільного захисту при ліквідації наслідків стихійних лих.

Рятувальні та невідкладні роботи при ліквідації наслідків стихійного лиха і аварії проводять рятувальні формування, а також залучене працездатне населення.

Роботи, які потребують спеціальної підготовки і залучення потужної техніки, виконують інженерні загони, створені на базі будівельних, ремонтно-будівельних, будівельно-монтажних, шляхово-будівельних організацій і відділів комунального господарства.

При стихійних лихах, які не є катастрофічними, всі заходи

організують керівники районів, об'єктів, населених пунктів і начальники загонів служби порятунку.

Начальник цивільного захисту (району, населеного пункту чи об'єкта) приймає рішення про проведення рятувальних і невідкладних робіт, в якому визначає: де зосередити основні зусилля, завдання рятувальних загонів і порядок введення їх на об'єкти робіт, початок і тривалість робіт, управління силами, які ведуть роботи, і порядок забезпечення їх дій.

Рятування людей і матеріальних цінностей є основним завданням при стихійних лихах. Послідовність виконання цього завдання залежить від характеру лиха, його наслідків, наявності та підготовки сил цивільного захисту, періоду року, стану погоди та інших факторів.

Тому захист населення, навколишнього природного середовища, промислових споруд і об'єктів від загрози природних факторів, а також надзвичайних ситуацій техногенного походження є важливим державним завданням [37].

ВИСНОВКИ

У процесі виконання дипломної кваліфікаційної роботи розв'язано актуальне завдання реалізації методу групового врахування аргументів. Проаналізовано процес редукції даних. Надано постановку задачі редукції даних. Показано, що етап редукції даних є одним з найбільш важливих етапів при синтезі розпізнавальних та діагностичних моделей, оскільки включення в модель малоінформативних ознак ускладнює процес побудови моделі, а також її подальше використання. Проаналізовано методи редукції даних. Визначено, що для редукції даних доцільно використовувати метод групового врахування аргументів. Досліджено метод групового врахування аргументів.

Досліджено та проаналізовано вимоги до програмного забезпечення для реалізації методу групового врахування аргументів. Для розробки програмного забезпечення обрано мову пакету Matlab, яка є зручною для створення відповідної програми та містить набір вбудованих бібліотек, що спрощують реалізацію математичних обчислень. Розроблено структурну схему програми. Визначено, що основними модулями розробленого програмного забезпечення є такі:

- Main.m, що містить первісну генерацію пар ознак, перерахунок порогового значення для відбору груп ознак та виведення кінцевих наборів ознак і спільних ознак у них;
- genData.m – відповідає за генерацію вхідних параметрів;
- regress.m – виконує регресійний аналіз для знаходження коефіцієнтів лінійної регресії в лінійних апроксимаціях функціональної залежності;
- repair.m – виконує формування нових груп ознак на кожній ітерації;
- getBestPairs.m – сортує групи ознак за величиною помилки апроксимації багатомірної моделі лінійною залежністю, сортує ознаки всередині кожного набору та вилучає повтори в них;

– gui.m – забезпечує графічний інтерфейс.

Наукова новизна роботи полягає у тому, що запропоновано модифікацію методу групового врахування аргументів, в якій, на відміну від існуючого методу, застосовується модифікований критерій селекції, використання якого передбачає, що на кожному наступному ряді селекції значення порогу поступово зменшується. Це дозволяє скоротити кількість математичних обчислень та прискорити збіжність алгоритму до оптимальних груп інформативних ознак.

Практичне значення отриманих у роботі результатів полягає у тому, що розроблено програмне забезпечення, що реалізує метод групового врахування аргументів для редукції даних.

Виконано тестування розробленого програмного забезпечення, що реалізує метод групового врахування аргументів. Результати тестування підтвердили, що програма може використовуватися за призначенням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Субботін С.О. Інтелектуальні системи : навчальний посібник / С. О. Субботін, А. О. Олійник ; під заг. ред. проф. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2013. – 216 с.
2. Encyclopedia of artificial intelligence / Eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. – New York : Information Science Reference, 2009. – Vol. 1-3. – 1677 p.
3. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. / С.О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с.
4. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С.О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.
5. Олійник А. О. Еволюційні обчислення та програмування: Навчальний посібник / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2010. – 324 с.
6. Ивахненко А.Г. Самоорганизация прогнозирующих моделей / А.Г. Ивахненко, Й.А.Мюллер. – К.: Техніка, 1985. – 223 с.
7. Ивахненко А.Г. Индуктивный метод самоорганизации моделей сложных систем / А.Г. Ивахненко. – К.: Наукова думка, 1981 — 296 с.
8. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами / А.Г. Ивахненко. – К.: Техніка, 1975, 312 с.
9. Ivakhnenko, A.G. Parametric and Nonparametric Selection Procedures in Experimental Systems Analysis / A.G. Ivakhnenko, J.A. Muller // Systems Analysis, Modeling and Simulation. – 1992. – № 9. – 157-175 p.
10. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиадвигателей : Монография / А. В. Богуслаев, Ал. А. Олейник, Ан. А. Олейник и др. ; под

ред. Д. В. Павленко, С. А. Субботина. – Запорожье : ОАО «Мотор Сич», 2009. – 468 с.

11. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2006. – 1408 с.

12. Зайченко Ю. П. Основы проектування інтелектуальних систем : навчальний посібник / Ю. П. Зайченко. – К.: Слово, 2004.– 352 с.

13. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М. : Вильямс, 2005. – 864 с.

14. Олійник А. О. Інтелектуальний аналіз даних: навчальний посібник / А. О. Олійник, С. О. Субботін, О.О. Олійник. – Запоріжжя : ЗНТУ, 2011. – 271 с.

15. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP: Уч. пос. / А. А. Барсегян. – СПб : BHV, 2007. – 384 с.

16. Брянцев И. Н. Data Mining. Теория и практика / И. Н. Брянцев. – М. : БДЦ-Пресс, 2006. – 208 с.

17. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов : монография / [С. А. Субботин, Ан.А. Олейник, Е. А. Гофман, С. А. Зайцев, Ал.А. Олейник]; под ред. С. А. Субботина. – Харьков: Компания СМІТ, 2012. – 318 с.

18. Рідкокаша А.А. Основы систем штучного інтелекту : . навчальний посібник / А.А. Рідкокаша, К.К. Голдер. – Черкаси, "Відлуння–Плюс", 2002.– 240 с.

19. Bentley J. L. Writing Efficient Programs (Prentice-Hall Software Series) [Text] / J. L. Bentley. – Englewood Cliffs, NJ : Prentice Hall, 1982. – 170 p.

20. Кнут Д. Искусство программирования [Текст] / Д. Кнут. — 3-е изд. — М.: Вильямс. — Т. 2, 2000 — 832 с.

21. Потемкин В. Г. Система Matlab. Справочное пособие / В. Г. Потемкин. – М.: Диалог-МИФИ, 1997. – 350 с.
22. Смоленцев Н. К. Создание Windows-приложений с использованием математических процедур Matlab / Н. К. Смоленцев. – М.: ДМК-Пресс, 2008. – 456 с.
23. Экономика и организация производства программных продуктов [Электрон. ресурс]. – 2019. – Режим доступа: <http://megaobzor.net/e-book/book.html>.
24. Депутат О. П. Цивільна оборона. Навчальний посібник. / О. П. Депутат, І. В. Коваленко, І. С. Мужик. – Львів.: Афіша, 2001. – 336 с.
25. Остапенко, В.В. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проєктування» і 7.091501 «Комп'ютерні та інтелектуальні системи і мережі» всіх форм навчання [Текст] / В.В. Остапенко, Е.М. Касьян, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2009. – 22 с.
26. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми [Текст]. – Харків : Форт, 2018. – 48 с.
27. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги [Текст]. – Київ : Держспоживстандарт України, 2015. – 15 с.
28. Правила улаштування електроустановок [Текст] : ПУЕ-2017. – На заміну ПУЕ-86 ; чинний з 2017-08-21. – К. : Міненерговугілля України, 2017. – 617 с.
29. ДСТУ 7237:2011 Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту [Текст]. – Київ : Держспоживстандарт України, 2011. – 10 с.

30. Охорона праці в галузі. Навчальний посібник [Текст] / П. Атаманчук, В. Мендерецький, О. Панчук, Р. Билык. – К. : Центр навчальної літератури, 2017. – 322 с.
31. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник [Текст] : проблеми природного і штучного освітлення. – К. : Етіп, 2019. – 180 с.
32. Шмирко, В.І. Методичні вказівки до лабораторного заняття «Розрахунок загального рівномірного штучного освітлення виробничих приміщень» з дисципліни «Цивільний захист і охорона праці в галузі» [Текст] : для студентів усіх спеціальностей та форм навчання / В.І. Шмирко, О.В. Коробко, Ю.І. Троян. – Запоріжжя : каф. ОПіНС. НУ «Запорізька політехніка», 2020. – 36 с.
33. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [Текст]. – Харків : Форт, 2000. – 24 с.
34. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку [Текст]. – К. : МОЗ України, 1999. – 24 с.
35. ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою [Текст]. – Київ : Мінрегіонбуд України, 2016. – 66 с.
36. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – Київ : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.
37. Стеблюк, М. І. Цивільна оборона та цивільний захист [Текст] : навч. посіб. для вузів / М. І. Стеблюк. – К. : Знання, 2013. – 487 с.

ДОДАТОК А
Технічне завдання

Вступ

Програма призначена для реалізації методу групового врахування аргументів.

A.1 Підстави для розробки

Підставою для розробки та реалізації проєкту є завдання на дипломну роботу «Дослідження та програмна реалізація методу групового врахування аргументів», затверджене наказом по Національному університету «Запорізька політехніка».

A.2 Призначення програми

Програмне забезпечення призначено для реалізації методу групового врахування аргументів для редукції великих масивів даних.

A.3 Вимоги до програмного забезпечення

A.3.1 Вимоги до функціональних характеристик програмного забезпечення

Програмне забезпечення повинно виконувати такі функції:

- забезпечення файлового введення/виведення інформації;
- можливість додавання нових значень у вибірку;
- редукція даних за допомогою методу групового врахування аргументів;
- збереження і завантаження результатів редукції даних;
- підтримка візуального інтерфейсу користувача.

Інтерфейс користувача повинен забезпечити легку взаємодію користувача з програмою.

A.3.2 Вимоги до надійності програмного забезпечення

Програмне забезпечення може коректно працювати, перевіряти чи відповідають вхідні дані потрібному формату. У разі невідповідності даних програма повинна видавати повідомлення про це.

A.3.3 Вимоги до експлуатації

Програмне забезпечення може бути записано на флешносії. Експлуатація програмного забезпечення здійснюється згідно експлуатаційної документації. Така документація містить інформацію, яка є потрібною для засвоєння та експлуатації програмни.

A.3.4 Вимоги до складу й параметрів технічних засобів

Для роботи з програмою необхідно мати таке апаратне забезпечення:

- комп'ютер під керівництвом операційної системи та встановленим пакетом Matlab;
- процесор частотою 2 ГГц;
- місце на жорсткому диску – 2 Гб;
- відеокарта з оперативною пам'яттю 64 Мб;
- оперативна пам'ять – 2 Гб;
- маніпулятор типу миша.

A.3.5 Вимоги до інформаційної сумісності

Для функціонування програмного забезпечення необхідна операційна система.

A.4 Стадії та етапи створення програмного забезпечення

Програмне забезпечення повинно бути створено у 16-ти тижневий строк від часу отримання завдання.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програма може використовуватися для розв'язання практичних завдань редукції великих масивів даних за допомогою методу групового врахування аргументів.

Б.2 Функціональне призначення

Програмне забезпечення призначено для реалізації методу групового врахування аргументів для редукції великих масивів даних.

Б.3 Опис логічної структури

Основними модулями розробленого програмного забезпечення є такі:

- Main.m, що містить первісну генерацію пар ознак, перерахунок порогового значення для відбору груп ознак та виведення кінцевих наборів ознак і спільних ознак у них;
- genData.m – відповідає за генерацію вхідних параметрів;
- regress.m – виконує регресійний аналіз для знаходження коефіцієнтів лінійної регресії в лінійних апроксимаціях функціональної залежності;
- repair.m – виконує формування нових груп ознак на кожній ітерації;
- getBestPairs.m – сортує групи ознак за величиною помилки апроксимації багатомірної моделі лінійною залежністю, сортує ознаки всередині кожного набору та вилучає повтори в них;
- gui.m – забезпечує графічний інтерфейс.

Б.4 Використані технічні засоби

Для роботи з програмним забезпеченням може знадобитися таке апаратне забезпечення:

- ПК під керівництвом операційної системи та встановленим пакетом Matlab;
- процесор частотою 2 ГГц;
- місце на жорсткому диску – 2 Гб;
- відеокарта з оперативною пам'яттю 64 Мб;
- оперативна пам'ять – 2 Гб;
- маніпулятор типу миша.

Б.5 Завантаження програмного забезпечення

Запуск програми відбувається за допомогою файлу Main.fig, в результаті чого буде запущено середовище Matlab і відкритий графічний інтерфейс. Іншим способом є запуск програми з консолі за командою run ('Main'), але для цього попередньо необхідно змінити поточну директорію на ту, в якій зберігається програма.

Б.6 Вхідні дані

Вхідними даними для програми є простір ознак та відповідний до них вектор y , що на практиці можна отримати в результаті збору даних після виконання серії експериментів.

Б.7 Вихідні дані

Вихідними даними для програми є побудовані групи корельованих між собою ознак, що можна замінити однією ознакою для спрощення розмірності моделі функціональної залежності.

ДОДАТОК В
Текст програми

```
% модуль Main.m  
function [] = Main()  
  
clc;  
%clear;  
  
global x y;  
global N S;  
  
N=30;  
S=100;  
load nIter;  
  
load x;  
load y;  
  
% generate pairs  
idx = 0;  
for i=1:N  
    for j=(i+1):N  
        idx = idx+1;  
        pair(idx).member1 = x(:,i);  
        pair(idx).member2 = x(:,j);  
        pair(idx).group = [i j];
```

```

        pair(idx).a = rand(3);
        pair(idx).err = Inf;
    end
end

nPairs = idx;

coef = 1;
for iter=1:nIter
    [bestPairs, indexes] = getBestPairs(pair, nPairs);
    nBest = min(N, length(bestPairs));
    nBest = round(nBest/coef);
    %coef = coef+1;
    coef = coef*1.61803399;
    [pair, nPairs] = rePair(bestPairs, nBest);
end

%creating file
file = fopen('out.txt','w');
fprintf(file,'%s\r\n','Best pairs: ');

%disp 'Best pairs: ';
nBest = round(min(N, length(bestPairs))/coef);
total = [];
un = bestPairs(1).group;
for i=1:nBest

```

```

    str = mat2str(bestPairs(i).group);
    fprintf(file,'%s\r\n',str);
    un = intersect(un, bestPairs(i).group);
    total = [total bestPairs(i).group];
end
disp 'Union: '
save un

```

```

% модуль genData.m
function [] = genData()
global S N;
N = 30;
S = 100;

```

```

x = rand(S, N);
y = (x(:,4)).*(x(:,1)+2*x(:,2)+7*x(:,3));

```

```

save x;
save y;

```

```

% модуль regress.m
function [b,bint,r,rint,stats] = regress2(y,X,alpha)

```

```

if nargin < 2
    error('stats:regress:TooFewInputs', ...

```

```

        'REGRESS requires at least two input arguments.');"
elseif nargin == 2
    alpha = 0.05;
end

[n,ncolX] = size(X);
if ~isvector(y)
    error('stats:regress:InvalidData', 'Y must be a vector.');"
elseif numel(y) ~= n
    error('stats:regress:InvalidData', ...
        'The number of rows in Y must equal the number of rows in X.');"
end

wasnan = (isnan(y) | any(isnan(X),2));
havenans = any(wasnan);
if havenans
    y(wasnan) = [];
    X(wasnan,:) = [];
    n = length(y);
end

[Q,R,perm] = qr(X,0);
p = sum(abs(diag(R)) > max(n,ncolX)*eps(R(1)));
if p < ncolX
    warning('stats:regress:RankDefDesignMat', ...
        'X is rank deficient to within machine precision.');"
    R = R(1:p,1:p);
end

```

```

    Q = Q(:,1:p);
    perm = perm(1:p);
end

b = zeros(ncolX,1);
b(perm) = R \ (Q'*y);

if nargout >= 2

    RI = R\eye(p);
    nu = max(0,n-p);
    yhat = X*b;
    r = y-yhat;
    normr = norm(r);
    if nu ~= 0
        rmse = normr/sqrt(nu);
        tval = tinv((1-alpha/2),nu);
    else
        rmse = NaN;
        tval = 0;
    end
    s2 = rmse^2;
    se = zeros(ncolX,1);
    se(perm,:) = rmse*sqrt(sum(abs(RI).^2,2));
    bint = [b-tval*se, b+tval*se];
    if nargout >= 4
        hatdiag = sum(abs(Q).^2,2);

```

```

ok = ((1-hatdiag) > sqrt(eps(class(hatdiag))));
hatdiag(~ok) = 1;
if nu > 1
    denom = (nu-1) .* (1-hatdiag);
    sigmai = zeros(length(denom),1);
    sigmai(ok) = sqrt(max(0,(nu*s2/(nu-1)) - (r(ok) .^2 ./ denom(ok))));
    ser = sqrt(1-hatdiag) .* sigmai;
    ser(~ok) = Inf;
elseif nu == 1
    ser = sqrt(1-hatdiag) .* rmse;
    ser(~ok) = Inf;
else
    ser = rmse*ones(length(y),1);
end

rint = [(r-tval*ser) (r+tval*ser)];
end

if nargout == 5
    SSE = normr.^2;
    RSS = norm(yhat-mean(y))^2;
    TSS = norm(y-mean(y))^2;
    r2 = 1 - SSE/TSS;
    if p > 1
        F = (RSS/(p-1))/s2;
    else
        F = NaN;
    end
end

```

```

prob = 1 - fcdf(F,p-1,nu);
stats = [r2 F prob s2];

if ~any(all(X==1,1))

    b0 = R\((Q'*ones(n,1));
    if (sum(abs(1-X(:,perm)*b0))>n*sqrt(eps(class(X))))
        warning('stats:regress:NoConst',...
            ['R-square and the F statistic are not well-defined' ...
            ' unless X has a column of ones.\nType "help' ...
            ' regress" for more information.']);
    end
end
end

if havenans
    if nargout >= 3
        tmp = repmat(NaN,length(wasnan),1);
        tmp(~wasnan) = r;
        r = tmp;
        if nargout >= 4
            tmp = repmat(NaN,length(wasnan),2);
            tmp(~wasnan,:) = rint;
            rint = tmp;
        end
    end
end
end

```

end

% модуль rePair.m

function [newpair, nPairs] = rePair(pairs, howMany)

if (howMany < 2)

 newpair = pairs(1);

 nPairs = 1;

 return

end

N=howMany;

idx = 0;

for i=1:N

 for j=(i+1):N

 idx = idx+1;

 newpair(idx).member1 = pairs(i).f;

 newpair(idx).member2 = pairs(j).f;

 newpair(idx).group = unique([pairs(i).group pairs(j).group]);

 % remove this to save time

 newpair(idx).a = rand(3);

 newpair(idx).err = Inf;

 for k=1:(idx-1)

 if (isequal(newpair(k).group, newpair(idx).group))

 disp('DUPLICATE');

 idx = idx - 1;

 break;

 end

 end

end

```

end
nPairs = idx;
end

% модуль getBestPair.m
function [bestPairs indexes] = getBestPairs(pair, nPairs)
global x y N S;

% find a0, a1, a2 (regress coeff)
for i=1:nPairs
    x1 = pair(i).member1;
    x2 = pair(i).member2;
    x_regr = [x1 x2 ones(S,1)];

    a = regress(y, x_regr);
    pair(i).a = a;

    pair(i).f = a(1).*x1 + a(2).*x2 + a(3);

    pair(i).err = sum((y-pair(i).f).^2);
end

err = [pair(:).err];
[y_sorted, indexes] = sort(err);

for i=1:nPairs
    bestPairs(i) = pair(indexes(i));
end

end

```

```

strAll = "";
i = 1;
file=fopen('out.txt');
while 1
    tlinee = fgetl(file);
    if ~ischarr(tline), break, end
    str = sprintf('%s',tline);
    strAll{i} = str;
    i = i+1;
end
fclose(file);

```

```

set(handles.bestPairs,'String',strAll);
guidata(hObject, handles);

```

```

if (get (hOObject,'Value') == geet(hObject,'Max'))
    fl = 1;
else
    fl = 0;
end
save fl;

```

```

Main();
load fl;

```

```
if fl == 1;  
    load un;  
    s = mat2str(un);  
    h = msgbox(s,'Union','help');  
end
```

ДОДАТОК Д
Слайди презентації

Національний університет “Запорізька політехніка”
кафедра програмних засобів

Дипломна кваліфікаційна робота

**Дослідження та програмна реалізація методу
групового врахування аргументів**

Виконав
ст. гр. КНТ-219м

К. С. Біловоденко

Керівник
к.т.н., доцент

С. Ю. Скрупський

Рисунок Д.1 – Слайд № 1

Об'єкт дослідження – процес редукції даних.

Предмет дослідження – метод групового врахування аргументів.

Мета роботи – дослідження та розробка програмних засобів для реалізації методу групового врахування аргументів.

Рисунок Д.2 – Слайд № 2

Завдання роботи:

- аналіз методу групового врахування аргументів;
- розробка технічного завдання для реалізації програми;
- вибір середовища розробки програмного забезпечення;
- розробка модифікованого методу групового врахування аргументів;
- конструювання програмного забезпечення для редукції даних шляхом використання методу групового врахування аргументів;
- тестування розробленого програмного забезпечення.

3

Рисунок Д.3 – Слайд № 3

Порівняння методів редукції даних [1]

Критерій порівняння	Методи відбору ознак												
	Методи перебору					Евристичні методи			Методи ранжирування й кластеризації		Методи випадкового пошуку		
	Методи повного перебору			Методи скороченого перебору									
	Класичний повний перебір	Пошук у глибину	Пошук заширшки	Метод гілок і границь	МГВА	Метод послідовного додавання ознак	Метод послідовного видалення ознак	Метод почергового додавання та видалення ознак	Ранжирування ознак	Кластеризація ознак	Випадковий пошук з адаптацією	Еволюційний пошук	
1	Оптимальність пошуку (оптимальність знайдених рішень)	Оптимальний	Оптимальний	Оптимальний	Субоптимальний	Субоптимальний	Субоптимальний	Субоптимальний	Субоптимальний	Субоптимальний	Субоптимальний	Субоптимальний	
2	Кількість рішень на кожній ітерації	Одне	Одне	Одне	Одне	Деякі N	Одне	Одне	Одне	Одне	Одне	Деякі N	Деякі N
3	Детермінованість пошуку	Детермінований	Детермінований	Детермінований	Детермінований	Детермінований	Детермінований	Детермінований	Детермінований	Детермінований	Стохастичний	Стохастичний	
4	Теоретичне обґрунтування	Високе	Високе	Високе	Середнє	Середнє	Середнє	Середнє	Середнє	Середнє	Середнє	Низьке	Середнє
5	Відносна складність реалізації	Низька	Низька	Низька	Середня	Середня	Середня	Середня	Середня	Низька	Середня	Середня	Висока
6	Принцип формування початкового рішення (початкова точка пошуку)	Порожня множина	Порожня множина	Порожня множина	Порожня множина	Деякі (N) одноозначних наборів	Порожня множина	Множина, що містить всі ознаки вихідного набору	Порожня множина	Порожня множина	Порожня множина	Деякі (N) випадково згенерованих або отриманих за певними правилами наборів	Деякі (N) випадково згенерованих або отриманих за певними правилами наборів
7	Критерії оцінювання інформативності ознак	Групові	Групові	Групові	Групові	Групові	Групові	Групові	Групові	Індивідуальні	Метрики відстані	Групові	Групові
8	Обчислювальна складність	$O(2^N)$	$O(2^N)$	$O(2^N)$	$O(L^N)$	$O(NL^N)$	$O(L^N)$	$O(L^N)$	$O(L^N)$	$O(L)$	$O(L^2)$	$O(NLT)$	$O(NLT)$

[1] Олійник А.О. Інтелектуальний аналіз даних : Навчальний посібник / А.О. Олійник, С.О.Субботін, О.О. Олійник. – Запоріжжя : ЗНТУ, 2012. – 271 с.

4

Рисунок Д.4 – Слайд № 4

Модифікований метод групового врахування аргументів

У запропонованій модифікації МГВА використовується модифікований критерій селекції, використання якого передбачає, що на кожному наступному селекційному ряді значення порогу зменшується за наступним виразом:

$$N_{i+1} = N_i / coef,$$

де N_i – значення порогу на поточному селекційному ряді;
 $coef$ – коефіцієнт зменшення порогу. Коефіцієнт $coef > 1$ задається користувачем. При цьому занадто великі значення цього коефіцієнту призведуть до швидкої зміни порогу до дуже малого числа, внаслідок чого методу не вдасться побудувати модель з прийнятною точністю. Дуже малі значення цього коефіцієнту будуть наближати реалізацію модифікованого МГВА до звичайного МГВА.

5

Рисунок Д.5 – Слайд № 5

Аналіз вимог до програмного забезпечення

Функціональними вимогами до програмного продукту (ПП) є такі:

- забезпечення файлового введення/виведення інформації;
- можливість додавання нових значень у вибірку;
- редукція даних за допомогою методу групового врахування аргументів;
- збереження і завантаження результатів редукції даних;

наявність візуального інтерфейсу користувача.

6

Рисунок Д.6 – Слайд № 6

Вибір інструментарію розробки програмного забезпечення

Критерії порівняння	Мови програмування		
	Matlab	Python	C++
Наявність великої кількості бібліотек для математичних розрахунків	++	+	+
Наявність літератури та простота вивчення	++	++	++
Підтримка парадигми модульного програмування	++	+	—
Продуктивність роботи	++	++	+
Можливість паралельного програмування	++	+	+
Простота розробки інтерфейсу користувача	++	±	+

7

Рисунок Д.7 – Слайд № 7

Структурна схема програми



8

Рисунок Д.8 – Слайд № 8

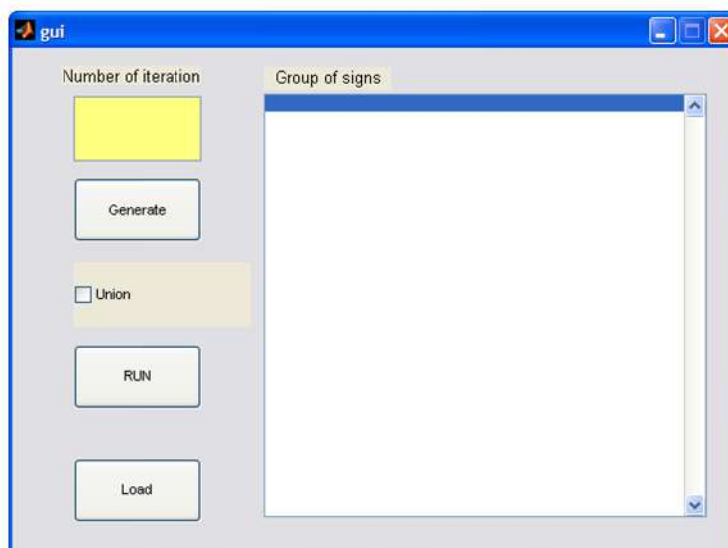
Схема функціонування програми



9

Рисунок Д.9 – Слайд № 9

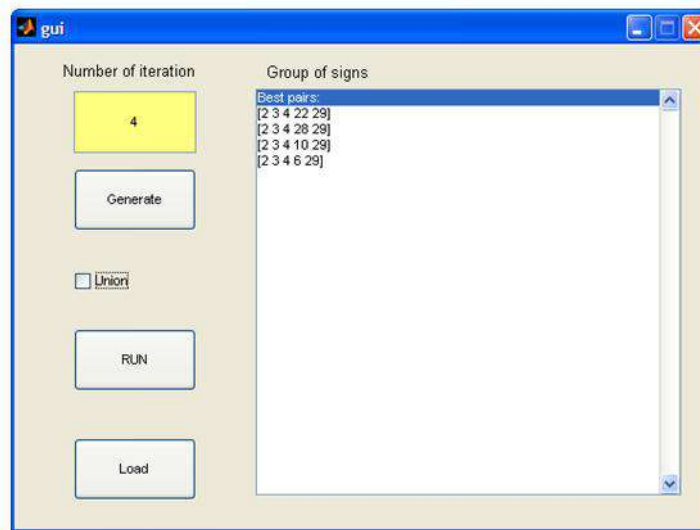
Приклад роботи з програмою. Головне вікно програми



10

Рисунок Д.10 – Слайд № 10

Приклад роботи з програмою. Результати роботи програми



11

Рисунок Д.11 – Слайд № 11

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було досліджено та програмно реалізовано метод групового врахування аргументів.

Роз'язано такі завдання:

- аналіз методу групового врахування аргументів;
- розробка технічного завдання для реалізації програми;
- вибір середовища розробки програмного забезпечення;
- розробка модифікованого методу групового врахування аргументів;
- конструювання програмного забезпечення для редукції даних шляхом використання методу групового врахування аргументів;
- тестування розробленого програмного забезпечення.

12

Рисунок Д.12 – Слайд № 12

Висновки

Наукова новизна роботи полягає у тому, що запропоновано модифікацію методу групового врахування аргументів, в якій, на відміну від існуючого методу, застосовується модифікований критерій селекції, використання якого передбачає, що на кожному наступному ряді селекції значення порогу поступово зменшується. Це дозволяє скоротити кількість математичних обчислень та прискорити збіжність алгоритму до оптимальних груп інформативних ознак.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення, що реалізує метод групового врахування аргументів для редукції даних.

Виконано тестування розробленого програмного забезпечення, що реалізує метод групового врахування аргументів. Результати тестування підтвердили, що програма може використовуватися за призначенням.

13

Рисунок Д.13 – Слайд № 13