ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

«ПОСТАНОВКА И РЕШЕНИЕ ДЕТЕРМИНИРОВАННЫХ ЗАДАЧ БЕЗУСЛОВНОЙ И УСЛОВНОЙ ОПТИМИЗАЦИИ»

ПО КУРСУ: «РАСПОЗНАВАНИЕ ОБРАЗОВ»

**ЛАБОРАТОРНАЯ РАБОТА №1**

Постановка и решение детерминированных задач безусловной и условной оптимизации

**Исходные данные на лабораторную работу**

Задана агрегированная модель бортовой системы (БС), представленная в виде таблицы состояний (таб. 1). Каждое из технических состояний (ТС) *S*1-*S*6 представляет собой неработоспособное состояние БС, вызванное одиночным отказом соответствующего функционального элемента. Модельные значения признаков  представляют собой вещественные числа.

Таблица 1. Состояния бортовой системы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ТС *Si* | Диагностические признаки π*j* | | | | | |
| π1 | π2 | π3 | π4 | π5 | π6 |
| *S*1 | 1.5 | 1.6 | 1.8 | -1.8 | 1.6 | 1.4 |
| *S*2 | 1.6 | 1.8 | 1.5 | -1.7 | 2 | 1.9 |
| *S*3 | 2 | 1.4 | 2 | -1.6 | 1.9 | 1.7 |
| *S*4 | 1.8 | 1.5 | 1.7 | -1.5 | 1.5 | 1.6 |
| *S*5 | 1.9 | 1.9 | 1.6 | -1.4 | 1.7 | 2.1 |
| *S*6 | 1.7 | 2.1 | 1.4 | -1.9 | 1.8 | 1.5 |

**Задание**.

1. Вычислить расстояния между классами ТС и сформировать матрицу межклассовых расстояний.

2. Исследовать модель и определить наиболее удаленные друг от друга классы ТС.

3. Определить интервалы разброса значений каждого из признаков в каждом из заданных технических состояний.

4. Построить таблицу состояний бортовой системы, в которой модельные значения диагностических признаков представлены в виде интервалов на вещественной числовой оси.

5. Сформировать множество информационных состояний и соответствующих им подмножеств допустимых для проверки признаков.

6. Сделать выводы по полученным результатам.

7. Оформить отчет о работе.

**Используемые формулы.**

1. Расстояние между классами:

– технические состояния, расстояния между которыми нужно рассчитать; – количество диагностических признаков; – модельное значение признака соответственно диагностического признака .

1. Определение наиболее удаленных классов:

*–* количество технических состояний.

1. Определение интервалов разброса значений

– текущее модельное значение признака, – модельное значение признака наиболее удаленного класса.

*–* левая и правая границы интервала разброса соответственно.

**Отчет по п.1**. Определение расстояний между классами и построение матрицы межклассовых расстояний

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| = | 0 | 0.748331 | 0.74162 | 0.5 | 0.974679 | 0.714143 |
| 0.748331 | 0 | 0.793725 | 0.74162 | 0.574456 | 0.591608 |
| 0.74162 | 0.793725 | 0 | 0.565685 | 0.812404 | 1.03923 |
| 0.5 | 0.74162 | 0.565685 | 0 | 0.69282 | 0.848528 |
| 0.974679 | 0.574456 | 0.812404 | 0.69282 | 0 | 0.860233 |
| 0.714143 | 0.591608 | 1.03923 | 0.848528 | 0.860233 | 0 |

**Отчет по п.2**. Определение наиболее удаленных классов

*Примечание. Здесь и далее нумерация классов начинается с нуля*

|  |  |
| --- | --- |
| Класс *Si* | Наиболее удаленный класс *Sf* |
| *S*0 | 4 |
| *S*1 | 2 |
| *S*2 | 5 |
| *S*3 | 5 |
| *S*4 | 0 |
| *S*5 | 32 |

**Отчет по п.3**. Определение интервалов разброса значений каждого из признаков в каждом из заданных технических состояний.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| = | 0.2 | 0.15 | 0.1 | 0.2 | 0.05 | 0.35 |
| 0.2 | 0.2 | 0.25 | 0.05 | 0.05 | 0.1 |
| 0.15 | 0.35 | 0.3 | 0.15 | 0.05 | 0.1 |
| 0.05 | 0.3 | 0.15 | 0.2 | 0.15 | 0.05 |
| 0.2 | 0.15 | 0.1 | 0.2 | 0.05 | 0.35 |
| 0.15 | 0.35 | 0.3 | 0.15 | 0.05 | 0.1 |

Левая граница интервалов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.3 | 1.45 | 1.7 | -2.0 | 1.55 | 1.05 |
| 1.4 | 1.6 | 1.25 | -1.75 | 1.95 | 1.8 |
| 1.85 | 1.05 | 1.7 | -1.75 | 1.85 | 1.6 |
| 1.75 | 1.2 | 1.55 | -1.7 | 1.35 | 1.55 |
| 1.7 | 1.75 | 1.5 | -1.6 | 1.65 | 1.75 |
| 1.55 | 1.75 | 1.1 | -2.05 | 1.75 | 1.4 |

Правая граница интервалов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.7 | 1.75 | 1.9 | -1.6 | 1.65 | 1.75 |
| 1.8 | 2.0 | 1.75 | -1.65 | 2.05 | 2.0 |
| 2.15 | 1.75 | 2.3 | -1.45 | 1.95 | 1.8 |
| 1.85 | 1.8 | 1.85 | -1.3 | 1.65 | 1.65 |
| 2.1 | 2.05 | 1.7 | -1.2 | 1.75 | 2.45 |
| 1.8 | 2.45 | 1.7 | -1.75 | 1.85 | 1.6 |

**Отчет по п.4.** Построение таблицы состояний бортовой системы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ТС *Si* | π1 | π2 | π3 | π4 | π5 | π6 |
| *S0* | [1.3, 1.7] | [1.45, 1.75] | [1.7, 1.9] | [-2.0, -1.6] | [1.55, 1.65] | [1.05, 1.75] |
| *S1* | [1.4, 1.8] | [1.6, 2.0] | [1.25, 1.75] | [-1.75, -1.65] | [1.95, 2.05] | [1.8, 2.0] |
| *S2* | [1.85, 2.15] | [1.05, 1.75] | [1.7, 2.3] | [-1.75, -1.45] | [1.85, 1.95] | [1.6, 1.8] |
| *S3* | [1.75, 1.85] | [1.2, 1.8] | [1.55, 1.85] | [-1.7, -1.3] | [1.35, 1.65] | [1.55, 1.65] |
| *S4* | [1.7, 2.1] | [1.75, 2.05] | [1.5, 1.7] | [-1.6, -1.2] | [1.65, 1.75] | [1.75, 2.45] |
| *S5* | [1.55, 1.85] | [1.75, 2.45] | [1.1, 1.7] | [-2.05, -1.75] | [1.75, 1.85] | [1.4, 1.6] |

**Отчет по п.5**. Формирование множества информационных состояний (ИС) и соответствующих им подмножеств допустимых для проверки признаков

проверка π1

проверка π2

проверка π3

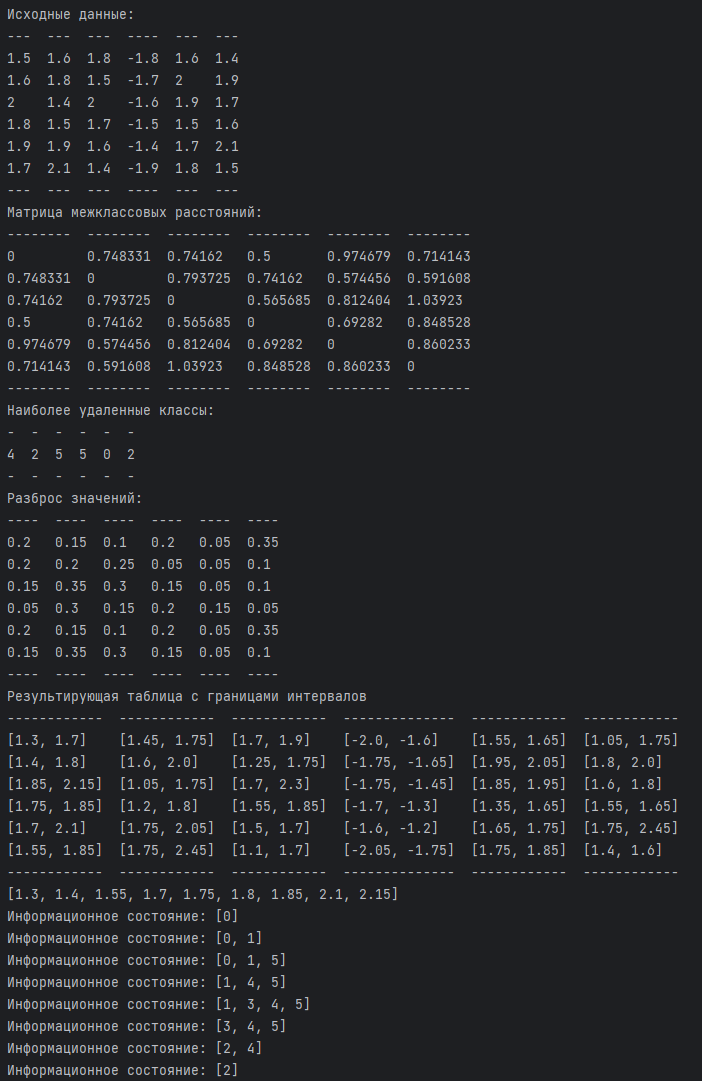
проверка π4

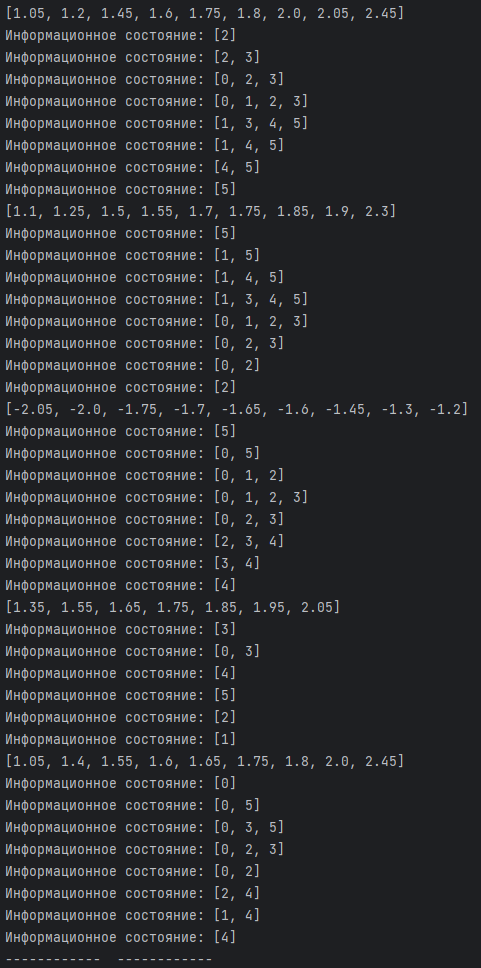
проверка π5

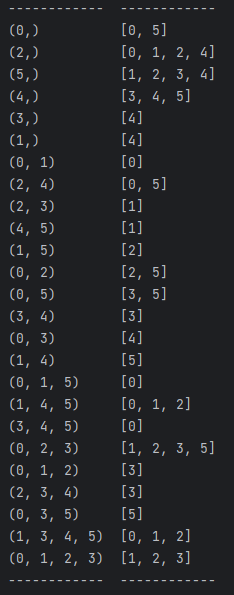
проверка π6

| ИС *Rk* | П*k* |  | ИС *Rk* | П*k* |
| --- | --- | --- | --- | --- |
| R7  {S0,S1} | Π1 |  | R23  {S3,S4,S5} | Π1 |
| R8  {S0,S2} | П3, П6 |  | R24  {S0,S1,S2,S3} | П2, П3, П4­ |
| R9  {S0,S3} | Π5 |  | R25  {S1,S3,S4,S5} | П1, П2, П3­ |
| R10  {S0,S5} | П4, П6 |  | R26  {S0,S1,S2,S3,S4,S5} | П1, П2, П3­, П4, П5, П6 |
| R11  {S1,S4} | Π6 |  |  |  |
| R12  {S1,S5} | Π3 |  |  |  |
| R13  {S2,S3} | Π2 |  |  |  |
| R14  {S2,S4} | П1, П6 |  |  |  |
| R15  {S3,S4} | Π4 |  |  |  |
| R16  {S4,S5} | Π1 |  |  |  |
| R17  {S0,S1,S2} | Π4 |  |  |  |
| R18  {S0,S1,S5} | Π1 |  |  |  |
| R19  {S0,S2,S3} | П2, П3, П4­, П6 |  |  |  |
| R20  {S0,S3,S5} | П6 |  |  |  |
| R21  {S1,S4,S5} | П1, П2, П3­ |  |  |  |
| R22  {S2,S3,S4} | П4 |  |  |  |

Описанные выше результаты были получены с помощью анализа результатов работы скрипта Python (Приложение А):

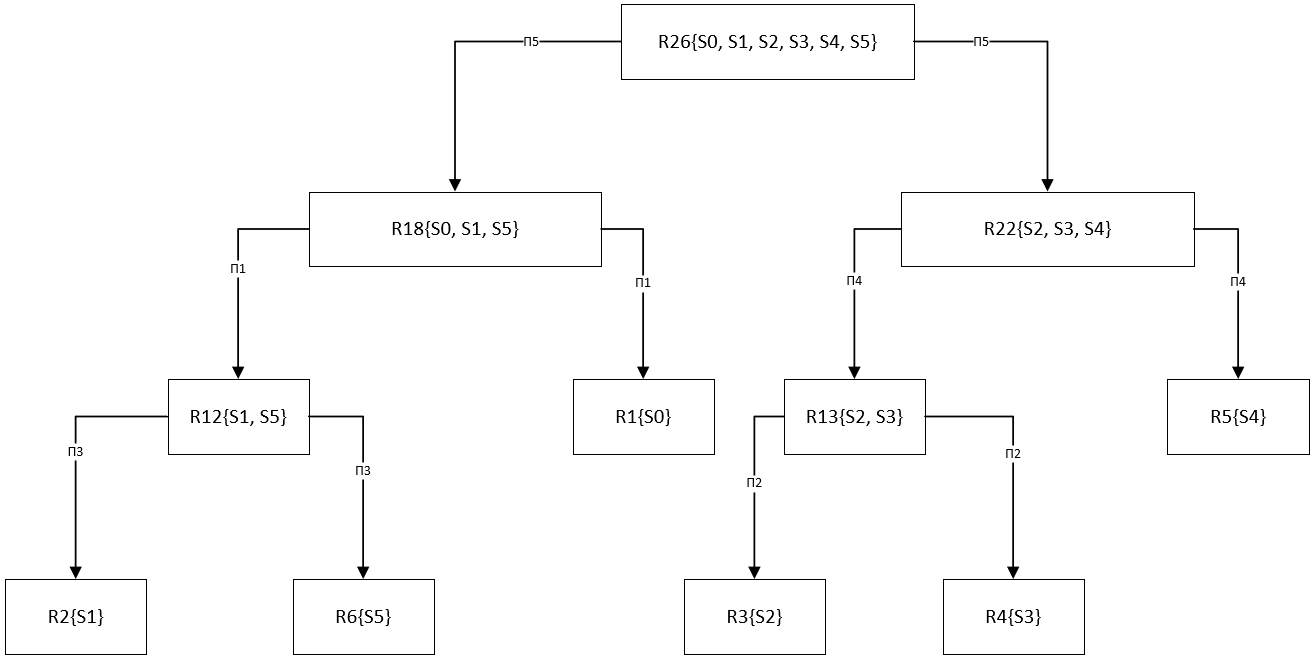






**Отчет по п.6**. Выводы по работе

В ходе выполнения лабораторной работы было получено множество информационных состояний и соответствующих им подмножеств допустимых для проверки признаков, по результатам которых можно построить график системы диагностирования:



Также в ходе работы был написан скрипт Python, который выполняет вышеуказанные действия.

Приложение А. Листинг программы.

**Скрипт Python.**

import math  
import matplotlib.pyplot as plt  
  
import tabulate  
  
def load\_data(file\_name):  
 with open(file\_name) as file:  
 numbers = []  
 for line in file:  
 temp\_line = []  
 for elem in line.rsplit():  
 temp\_line.append(float(elem))  
 numbers.append(temp\_line)  
 return numbers  
  
def get\_distance(states):  
 numbers = []  
 for i in range(len(states)):  
 temp\_numbers = []  
 for j in range(len(states[0])):  
 summary = 0  
 for i\_state in range(len(states)):  
 summary += pow(states[i][i\_state] - states[j][i\_state], 2)  
 temp\_numbers.append(math.sqrt(summary))  
 numbers.append(temp\_numbers)  
 return numbers  
  
def get\_far\_classes(distances):  
 numbers = []  
 for line in distances:  
 numbers.append(line.index(max(line)))  
 return numbers  
  
def get\_incs(states, far):  
 numbers = []  
 for i in range(len(states)):  
 temp\_numbers = []  
 for j in range(len(states)):  
 temp\_numbers.append(0.5 \* abs(states[i][j] - states[far[i]][j]))  
 numbers.append(temp\_numbers)  
 return numbers  
  
def get\_left\_right(states, incs):  
 numbers = []  
 for i in range(len(states)):  
 temp\_numbers = []  
 for j in range(len(states)):  
 temp\_numbers.append([round(states[i][j] - incs[i][j], 2), round(states[i][j] + incs[i][j], 2)])  
 numbers.append(temp\_numbers)  
 return numbers  
  
s = load\_data("LAB-1\_Data.txt")  
print("Исходные данные:")  
print(tabulate.tabulate(s))  
  
d = get\_distance(s)  
print("Матрица межклассовых расстояний:")  
print(tabulate.tabulate(d))  
  
f = get\_far\_classes(d)  
print("Наиболее удаленные классы:")  
print(tabulate.tabulate([f]))  
  
inc = get\_incs(s, f)  
print("Разброс значений:")  
print(tabulate.tabulate(inc))  
  
  
lr = get\_left\_right(s, inc)  
print("Результирующая таблица с границами интервалов")  
print(tabulate.tabulate(lr))  
  
  
plots\_y = []  
for i in range(len(s)):  
 plots\_y.append([i, i])  
  
info\_states = {}  
  
# для каждого признака определяем информационное состояние  
for k in range(len(s)):  
 plots\_x = []  
 points = []  
 for j in range(len(s)):  
 plots\_x.append([lr[j][k][0], lr[j][k][1]])  
 points.append(lr[j][k][0])  
 points.append(lr[j][k][1])  
 points = set(points)  
 points = list(points)  
 points.sort()  
 print(points)  
  
 # находим признаки  
 for i in range(len(points) - 1):  
 mid = 0.5 \* (points[i + 1] - points[i]) + points[i]  
 stations = []  
 for j in range(len(plots\_x)):  
 if plots\_x[j][0] <= mid <= plots\_x[j][1]:  
 stations.append(j)  
 print("Информационное состояние:", stations)  
 if tuple(stations) in info\_states.keys():  
 temp = info\_states.get(tuple(stations))  
 temp.append(k)  
 info\_states[tuple(stations)] = temp  
 else:  
 info\_states[tuple(stations)] = [k]  
 # графики  
 for i in range(len(s)):  
 plt.plot(plots\_x[i], plots\_y[i], marker='o')  
 plt.title("Признак №" + str(k + 1))  
 plt.grid(True)  
 plt.show()  
  
# сортировка полученных состояний и вывод в таблице  
keys = list(info\_states.keys())  
keys.sort(key=len)  
list\_info\_states = [[i, info\_states[i]] for i in keys]  
print(tabulate.tabulate(list\_info\_states))

**Файл «LAB-1\_Data.txt»:**

1.5 1.6 1.8 -1.8 1.6 1.4  
1.6 1.8 1.5 -1.7 2 1.9  
2 1.4 2 -1.6 1.9 1.7  
1.8 1.5 1.7 -1.5 1.5 1.6  
1.9 1.9 1.6 -1.4 1.7 2.1  
1.7 2.1 1.4 -1.9 1.8 1.5