Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра прикладной информатики

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 5**

Интеллектуальный анализ данных

наименование дисциплины

Нечеткие модели знаний (вариант 17)

тема (вариант)

Преподаватель \_\_\_\_\_\_\_\_\_\_ Т.Н. Сизова

подпись, дата инициалы, фамилия

Студент КИ22-21б 032213957 \_\_\_\_\_\_\_\_\_\_ А.С. Полешко

номер группы номер зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc192003516)

[Ход работы 4](#_Toc192003517)

[ЗАКЛЮЧЕНИЕ 4](#_Toc192003518)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc192003519)

# **ВВЕДЕНИЕ**

Цель работы:получение знаний, умений и навыков по созданию базы знаний, представляющей собой нечеткие модели знаний.

# **Ход работы**

Описание задачи: дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества и определить степень принадлежности одного элемента множеству D, используя максиминный метод (рисунок 1).

A graph of a line

AI-generated content may be incorrect.

Рисунок – Функции принадлежности для нечетких множеств A, B, C

Описание процесса решения задачи. Для построения функции принадлежности нового множества необходимо:

1) Определить последовательность выполнения операций в формуле.

2) Построить на отдельных графиках промежуточные множества, согласно определенной последовательности действий. Свести промежуточные множества на одном графике и определить итоговую функцию принадлежности.

3) Используя определенный в задаче метод, определить аналитически степень принадлежности элемента, входящего в носитель итогового множества.

4) Проверить аналитические вычисления по построенному графику функции принадлежности.

Решение.

1) Множество значит, последовательность операций будет следующей: , , , , .

2) Построим согласно этой последовательности операций графики функций принадлежности (рисунок 2-6).

A graph with a line

AI-generated content may be incorrect.

Рисунок – Функция принадлежности для множества

A graph with a line going up

AI-generated content may be incorrect.

Рисунок – Функция принадлежности для множества

A graph with a line

AI-generated content may be incorrect.

Рисунок 4 – Функция принадлежности для множества

A green line graph on a grid

AI-generated content may be incorrect.

Рисунок 5 – Функция принадлежности для множества

A graph with a line drawn on it

AI-generated content may be incorrect.

Рисунок 6 – Функция принадлежности для множества

3) Носитель множества D состоит из элементов из интервала (0,14). Выберем элемент 8.

4) Проверим аналитическое решение по графику (рисунок 7).

A graph with a red line

AI-generated content may be incorrect.

Рисунок – Проверка аналитического решения

**Нечеткий вывод**

Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи оценки сложности SQL-запроса (учитывая количество JOIN, подзапросов и агрегатных функций), проверить её на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).

Для реализации логического вывода необходимо выполнить следующее:

1) Сформулировать на естественном языке в виде предложений «Если … , то» закономерности предметной области.

2) Выделить из этих предложений лингвистические переменные, их значения (построить их функции принадлежности), высказывания различных видов, формализовать нечеткие правила.

3) Проверить полученную базу знаний на полноту.

4) Провести фаззификацию (входные данные выбираем случайным образом).

5) Провести аккумуляцию.

6) Провести дефаззицикацию.

**Решение.** Пусть перед нами стоит следующая задача: оценить сложность SQL-запроса на основе анализа количества JOIN, агрегатных функций и подзапросов.

1) Предложения, описывающие задачу, следующие:

– Если количество JOIN низкое и агрегатных функций мало и подзапросов нет, то сложность низкая;

– Если количество JOIN низкое и агрегатных функций мало и подзапросов немного, то сложность низкая;

– Если количество JOIN низкое и агрегатных функций мало и подзапросов много, то сложность средняя;

– Если количество JOIN среднее и агрегатных функций умеренно и подзапросов нет, то сложность низкая;

– Если количество JOIN среднее и агрегатных функций умеренно и подзапросов немного, то сложность средняя;

– Если количество JOIN среднее и агрегатных функций много и подзапросов много, то сложность высокая;

– Если количество JOIN высокое и агрегатных функций много и подзапросов много, то сложность высокая;

– И так далее для всех логичных комбинаций.

2) Выделим из правил лингвистические переменные:

– = количество JOIN, T = {низкое, среднее, высокое}, X = [0, 10], G = {очень низкое}, M – возведение в квадрат функции принадлежности .

– = количество агрегатных функций, T = {мало, умеренно, много}, X = [-10, 10], G = {очень много, немного}, M – очень много – возведение в квадрат функции принадлежности ., немного – извлечение квадратного корня из функции принадлежности .

– = количество подзапросов, T = {нет, немного, много}, X = [0, 5], G = {очень много, почти нет}, M – очень много – возведение в квадрат функции принадлежности , почти нет – извлечение квадратного корня из функции принадлежности .

– = сложность SQL-запроса, T = {низкая, средняя, высокая}, X = [0, 100], G = {очень высокая, слегка высокая}, M: очень высокая – возведение в квадрат функции принадлежности , слегка высокая – извлечение квадратного корня из функции принадлежности .

С учетом выделенных лингвистических переменных, нечеткие правила следующие:

– Если количество JOIN = «очень низкое» и количество агрегатных функций = «мало», и подзапросов = «нет», то сложность SQL-запроса = «низкая»;

– Если количество JOIN = «среднее» и агрегатов = «мало», и подзапросов = «немного», то сложность SQL-запроса = «средняя»;

– Если количество JOIN = «высокое» и агрегатов = «много», и подзапросов = «много», то сложность SQL-запроса = «высокая»;

– Если количество JOIN = «среднее» и агрегатов = «умеренно», и подзапросов = «немного», то сложность SQL-запроса = «средняя»;

– Если количество JOIN = «высокое» и агрегатов = «умеренно», и подзапросов = «немного», то сложность SQL-запроса = «высокая»;

– Если количество JOIN = «низкое» и агрегатов = «много», и подзапросов = «нет», то сложность SQL-запроса = «средняя»;

– Если количество JOIN = «среднее» и агрегатов = «много», и подзапросов = «много», то сложность SQL-запроса = «высокая»;

– Если количество JOIN = «низкое» и агрегатов = «мало», и подзапросов = «много», то сложность SQL-запроса = «средняя»;

– Если количество JOIN = «высокое» и агрегатов = «много», и подзапросов = «немного», то сложность SQL-запроса = «высокая»;

– Если количество JOIN = «очень низкое» и агрегатов = «мало», и подзапросов = «немного», то сложность SQL-запроса = «низкая».

3) Проверим полученную базу на полноту:

– Существует хотя бы одно правило для каждого терма выходной переменной («низкая», «средняя», «высокая» сложность);

– Для каждого терма входных переменных (JOIN, агрегаты, подзапросы) присутствует хотя бы одно правило, в котором он участвует.

Полученная база нечетких правил полная.

4) Пусть имеется SQL-запрос, содержащий 3 оператора JOIN, 2 агрегатные функции и 2 подзапроса. Блок «Фаззификация» вычисляет их степени принадлежности нечетким множествам и . Далее вычисляются степени выполнения условий отдельных правил (агрегация условий). Далее определяются модифицированные функции принадлежности заключений отдельных правил.

Методом центра тяжести в блоке «Дефаззификация» было определено, что для данных входных значений сложность SQL-запроса составляет 0.36 – т.е. средняя сложность.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы были получены знания, умения и навыки по созданию базы знаний, представляющей собой нечеткие модели знаний.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Электронный курс образовательной программы по дисциплине «Интеллектуальный анализ данных» // Система электронного обучения Сибирского Федерального университета: [сайт]. – Красноярск, 2010 – . – URL: <https://e.sfu-kras.ru/course/view.php?id=1109> (дата обращения: 11.03.2025). – Режим доступа: для авториз. пользователей. – Текст: электронный.