МИНИСТЕРСТВО НАУК И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ИМЕНИ И.М. ГУБКИНА

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

**ОТЧЕТ**

**Лабораторной работе №3**

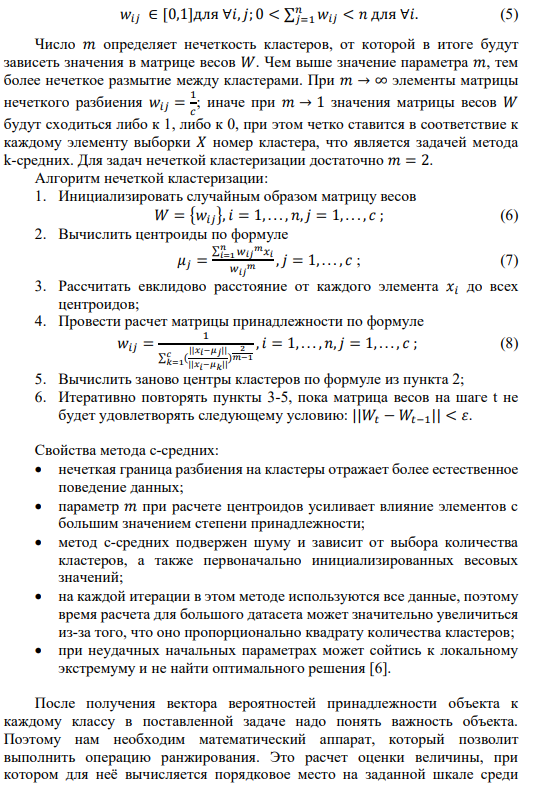
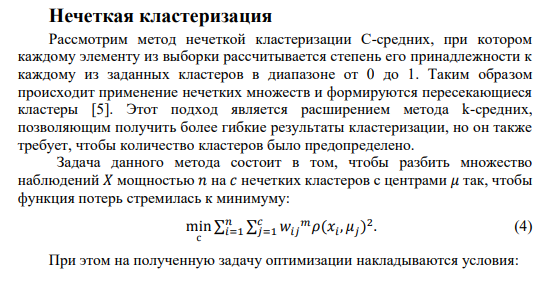
«Кластеризация методом с-средних»

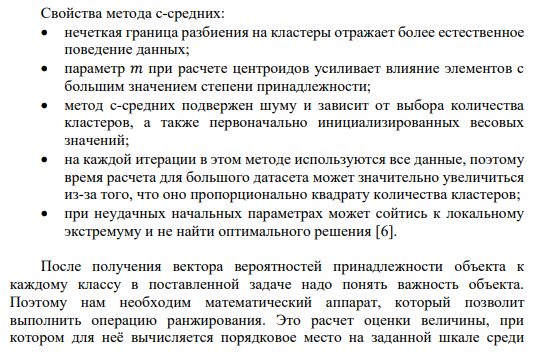
По дисциплине «Теория принятия решения. Нечеткие модели»

Выполнил работу студент группы АС-21-05

Алешко Альберт Алексеевич

Проверил Манзюк Николай Юрьевич

Москва, 202



Листинг кода:

import numpy as np

import skfuzzy as fuzz

import matplotlib.pyplot as plt

import matplotlib.colors as mcolors

def initialize\_membership\_matrix(n\_samples, n\_clusters):

    """Инициализация матрицы принадлежности случайными значениями"""

    U = np.random.dirichlet(np.ones(n\_clusters), size=n\_samples).T

    return U

def compute\_centroids(data, U, m):

    """Вычисление центроидов кластеров"""

    um = U \*\* m

    return (um @ data) / np.sum(um, axis=1, keepdims=True)

def update\_membership\_matrix(data, centroids, m):

    """Обновление матрицы принадлежности"""

    distances = np.linalg.norm(data[:, np.newaxis, :] - centroids, axis=2)

    distances = np.fmax(distances, np.finfo(np.float64).eps)

    inv\_distances = distances \*\* (-2 / (m - 1))

    return inv\_distances / np.sum(inv\_distances, axis=1, keepdims=True)

def fuzzy\_c\_means(data, n\_clusters=3, m=2, error=1e-5, max\_iter=1000):

    """Алгоритм нечеткой кластеризации C-средних"""

    n\_samples = data.shape[0]

    U = initialize\_membership\_matrix(n\_samples, n\_clusters)

    for \_ in range(max\_iter):

        U\_old = U.copy()

        centroids = compute\_centroids(data, U, m)

        U = update\_membership\_matrix(data, centroids, m).T

        if np.linalg.norm(U - U\_old) < error:

            break

    return centroids, U

def defuzzification(U):

    """Ранжирование объектов по важности через дефаззификацию"""

    return np.argmax(U, axis=0)

# Генерация случайных данных

np.random.seed(0)

f\_c = np.random.randn(1000, 2)

s\_c = np.random.rand(1000, 2) \* 0.2 + 0.5

t\_c = np.random.randn(1000, 2) \* 0.7 - 0.2

data = np.vstack((f\_c, s\_c, t\_c))

# Параметры кластеризации

num\_clusters = 3

fuzziness = 2

# Применение FCM

centroids, U = fuzzy\_c\_means(data, n\_clusters=num\_clusters, m=fuzziness)

def blend\_colors(memberships, color\_list):

    """Смешение цветов на основе степеней принадлежности"""

    # Преобразуем названия цветов в RGB

    colors\_rgb = [mcolors.to\_rgb(color) for color in color\_list]

    # Умножаем каждый цвет на степень принадлежности и суммируем

    blended = np.sum([np.array(c) \* m for c, m in zip(colors\_rgb, memberships)], axis=0)

    # Нормализуем результат

    return np.clip(blended, 0, 1)

# Визуализация результатов

fig, ax = plt.subplots(figsize=(8, 6))

colors = ['blue', 'green', 'purple', 'orange', 'brown', 'pink', 'gray'][:num\_clusters]

norm = mcolors.Normalize(vmin=0, vmax=1)

# Создание смешанных цветов для точек

point\_colors = [blend\_colors(U[:, i], colors) for i in range(data.shape[0])]

# Отображение данных и центроидов

ax.scatter(data[:, 0], data[:, 1], color=point\_colors, alpha=0.8, label='Данные')

ax.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='X', s=200, label='Центры кластеров')

# Настройка графика

ax.set\_title('Нечеткая кластеризация C-средних')

ax.set\_xlabel('Признак 1')

ax.set\_ylabel('Признак 2')

ax.legend()

plt.show()

# Дефаззификация - определение основного кластера для каждой точки

clusters = defuzzification(U)

# Визуализация четкого распределения по кластерам

plt.figure(figsize=(8, 6))

scatter = plt.scatter(data[:, 0], data[:, 1], c=clusters, cmap='viridis', alpha=0.8)

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='X', s=200, label='Центры кластеров')

# Добавление легенды для кластеров

legend1 = plt.legend(\*scatter.legend\_elements(), title="Кластеры")

plt.gca().add\_artist(legend1)

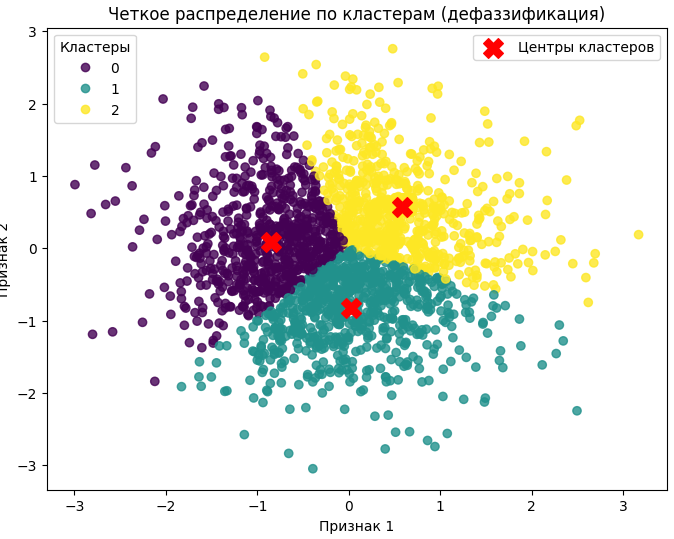
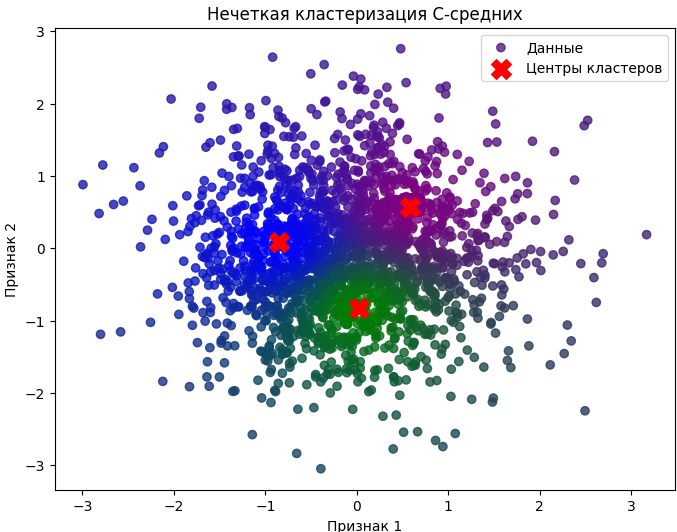
plt.title('Четкое распределение по кластерам (дефаззификация)')

plt.xlabel('Признак 1')

plt.ylabel('Признак 2')

plt.legend()

plt.show()

Результат:  
  
  
  
Вопросы:  
  
1. C-средних и метод локтя – в чем отличия?

C-средних выполняет кластеризацию, а метод локтя помогает выбрать подходящее значение k для C-средних. Метод локтя — это вспомогательный инструмент, который используется для улучшения результатов C-средних, но не является алгоритмом кластеризации сам по себе. Он может быть неточным и требует субъективной интерпретации, но часто дает хорошее начальное приближение для оптимального числа кластеров.  
  
2. Что такое дефаззификация и зачем она нужна (и нужна ли вообще)?

Дефаззификация — это процесс преобразования нечеткого множества или нечеткого числа в четкое значение. В контексте нечеткой логики и нечетких систем, мы часто работаем с нечеткими переменными, которые описываются функциями принадлежности, показывающими степень принадлежности к различным лингвистическим терминам (например, "низкий", "средний", "высокий"). Дефаззификация необходима, когда нам нужно получить конкретное числовое значение из нечеткого представления.

Зачем нужна дефаззификация?

Нечеткие системы часто используются для моделирования сложных систем, где существуют неопределенности и неточности. Результат работы нечеткой системы обычно представляет собой нечеткое множество, которое не всегда удобно использовать на практике. Дефаззификация позволяет перевести этот нечеткий результат в четкое значение, которое может быть использовано для управления, принятия решений или других действий в реальном мире. Например, нечеткая система управления может выдать нечеткий вывод "скорость должна быть примерно средней", который затем нужно преобразовать в конкретное значение скорости (например, 50 км/ч).

Нужна ли дефаззификация вообще?

Это зависит от задачи. В некоторых случаях дефаззификация необходима для взаимодействия нечеткой системы с реальным миром, где требуются четкие значения. В других случаях, нечеткий результат может быть достаточно информативным, и дефаззификация может быть излишней или даже вредной, поскольку может привести к потере информации. Например, если нечеткий вывод достаточно точно описывает желаемое состояние системы, то дополнительное преобразование в четкое значение может быть не нужно.

В общем, дефаззификация — важный, но не всегда обязательный шаг в работе с нечеткими системами. Выбор метода дефаззификации и необходимость ее применения должны определяться конкретной задачей и требуемой точностью результата.