ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 4

«Кластеризация массива посредством полного перебора всех комбинаций значений этого массива»

Выполнил работу

Конькова Нина

Академическая группа J3113

Принято

Старший преподаватель, Ходненко И. В.

Санкт-Петербург

2024

1. Введение

Цель работы: исследовать и реализовать алгоритм, выполняющий разбиение массива чисел на несколько кластеров с минимизацией суммы абсолютных отклонений (SAD) от среднего значения каждого кластера.

Задачи:

1. Изучить теоретические основы кластеризации и методов минимизации отклонений.
2. Разработать и реализовать алгоритм кластеризации, обеспечивающий разбиение массива на заданное количество кластеров.
3. Проанализировать производительность алгоритма с различными размерами входных данных и оценить соответствие теоретическим оценкам.
4. Теоретическая подготовка

В данной работе используются следующие понятия и методы:

**—** Кластеризация – процесс группировки элементов массива в подмножества (кластеры), для каждого из которых рассчитывается среднее значение.

**—** Сумма абсолютных отклонений (SAD) – критерий, используемый для оценки качества кластеризации. SAD определяет сумму абсолютных отклонений элементов от среднего значения в кластере.

**—** Рекурсивное разбиение – используется для генерации всех возможных вариантов разбиения массива на кластеры.

**—** Асимптотическая сложность алгоритма – оценка, используемая для анализа временной сложности алгоритма, зависящей от размера входного массива.

1. Реализация

В ходе выполнения работы был разработан алгоритм, который генерирует все возможные разбиения массива на заданное количество кластеров, затем для каждого разбиения вычисляется сумма абсолютных отклонений. Минимальная SAD выбирается как оптимальное разбиение.

Использованные функции:

**—** calcDev: вычисляет SAD для подмассива на основе разности элементов с их средним значением.

**—** genParts: рекурсивно генерирует все разбиения массива на подмассивы.

— findBest: Основная функция, обеспечивающая поиск оптимального разбиения на кластеры.

Основные этапы алгоритма:

1. Генерация всех разбиений массива.
2. Для каждого разбиения подсчет суммы отклонений от среднего.
3. Выбор минимального значения SAD среди всех разбиений.

Программный код реализован на языке C++ с использованием библиотеки vector для работы с динамическими массивами. Особенностью реализации является использование рекурсивного подхода для генерации разбиений, что может приводить к увеличению временной сложности для больших входных массивов.

1. Экспериментальная часть

Согласно требованиям моего варианта, на вход к моему алгоритму подаётся до 25 элементов. Теоретически заданная сложность задачи составляет O(2^N) и более. Для тестирования алгоритма была собрана статистика, приведенная в таблице №1

Таблица №1 - Подсчёт сложности реализованного алгоритма

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Размер входного набора | 1 | 5 | 10 | 15 | 20 | 25 |
| Время выполнения программы, мс | 0 | 0 | 0 | 38 | 1316 | 66556 |

График представляющий визуально удобный формат данных из таблицы №1 представлен на изображении №1.

Изображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описание

Изображение №1 - График работы алгоритма

Таблица №2 – Подсчёт памяти реализованного алгоритма

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Размер входного набора | 1 | 5 | 10 | 15 | 20 | 25 |
| Затраты по памяти, байты | 16 | 680 | 41040 | 1966200 | 83886240 | 3355443400 |

Анализ графика и таблицы: как видно из графика, время выполнения алгоритма экспоненциально возрастает с увеличением размера входного массива. Это подтверждает теоретическую оценку сложности алгоритма O(2^N), что свидетельствует о высокой сложности при увеличении числа элементов.

1. Заключение

В ходе выполнения работы был реализован алгоритм кластеризации массива с минимизацией суммы абсолютных отклонений. Цель работы была достигнута путем разработки и тестирования алгоритма на нескольких наборах данных различного размера. Полученные результаты подтвердили теоретическую оценку сложности, так как время выполнения росло экспоненциально с увеличением числа элементов массива.

Для дальнейших исследований рекомендуется оптимизация алгоритма за счет применения динамического программирования или использования жадных подходов для крупных массивов, а также разработка параллельной версии алгоритма для повышения производительности.

Приложения

ПРИЛОЖЕНИЕ А

Листинг кода файла main.cpp

#include <iostream>

#include <vector>

#include <cmath>

#include <limits>

#include <algorithm>

using namespace std;

// Функция для вычисления суммы абсолютных отклонений от среднего кластера

double calcDev(const vector<double>& arr, int start, int end) {

// Код функции

}

// Рекурсивная функция для генерации всех разбиений массива на подмассивы

void genParts(const vector<double>& arr, int index, vector<double>>& currPart,

vector<vector<vector<double>>>& allParts)

// Код функции

}

// Функция для нахождения лучшего разбиения массива на K кластеров

double findBest(vector<double> arr, int k) {

// Код функции

}

int main() {

// Код функции main

return 0;

}