

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образование «Полоцкий государственный университет имени
Евфросинии Полоцкой»

Факультет информационных технологий
Кафедра вычислительных систем и сетей

Лабораторная работа №1
по дисциплине: «Распределенные вычисления»
на тему: «Реализация алгоритма кластеризации методом К-средних на CPU.»

Выполнил

Студент гр. 23-ИТ-1
Вечерский М.И.

Проверил

Преподаватель кафедры ВСиС
Васильева Д.М.

Полоцк 2025

Вариант: 8

Цель работы: ознакомиться с алгоритмом кластеризации методом К-средних и реализовать данный алгоритм на CPU.

Описание проделанной работы: в ходе лабораторной работы был реализован и исследован алгоритм кластеризации k-means (метод k-средних) на языке C++ с использованием стандартной библиотеки. Целью работы было изучить принципы функционирования алгоритма, особенности его реализации и оценить производительность при обработке большого объёма данных.

Для эксперимента были заданы следующие параметры:

- количество точек - 80000;
- размерность пространства - 9;
- число кластеров - 10;
- максимальное число итераций - 100.

(update step) — для каждого кластера вычислялось среднее арифметическое координат всех точек, отнесенных к нему на предыдущем шаге; полученные значения становились новыми координатами соответствующих центроидов.

Указанные шаги последовательно повторялись в течение фиксированного числа итераций (100). Для пересчета положения центроидов была разработана специализированная функция, обеспечивающая суммирование координат точек каждого кластера с последующим делением на их количество.

С целью оценки эффективности реализации проводилось измерение времени выполнения алгоритма на центральном процессоре с использованием стандартной функции измерения времени. По завершении вычислений программа выводила суммарную длительность обработки всех итераций.

В результате выполнения лабораторной работы была получена работоспособная реализация алгоритма k-means, способная обрабатывать объемные наборы многомерных данных и демонстрирующая ключевые этапы процесса кластеризации. Практическая работа способствовала закреплению навыков оперирования многомерными данными, применения циклических конструкций, использования динамических структур данных и освоению базовых методик анализа производительности программного кода.

Листинг 1 – алгоритм k-средних

```
Point avg_point(const std::vector<Point>& points) {
    Point center;
    for (int i = 0; i < 9; ++i) {
        double sum = 0.0;
        for (const auto& point : points) {
            sum += point.coords[i];
        }
        center.coords[i] = sum / points.size();
    }
    return center;
}
```

Продолжение листинга 1

```
double distance(const Point& point, const Point& center) {
    double dist = 0.0;

    for (auto&& [p_coord, c_coord] : std::views::zip(point.coords,
center.coords)) {
        dist += (p_coord - c_coord) * (p_coord - c_coord);
    }

    return dist;
}

std::vector<Point> load_pointers(const std::string& path_to_dir)
{
    std::vector<Point> points;

    for (const auto& entry :
std::filesystem::directory_iterator(path_to_dir)) {
        if (entry.is_regular_file() && entry.path().extension()
== ".txt") {
            std::ifstream file(entry.path());

            std::string line;
            while (std::getline(file, line)) {
                std::istringstream iss(line);
                Point p;
                for (int i = 0; i < 9; ++i) {
                    iss >> p.coords[i];
                }
                points.push_back(p);
            }
        }
    }
    return points;
}

void save_all_results(const std::vector<Point>& points, const
std::vector<Point>& centers) {
    std::filesystem::create_directories("../Data");

    std::ofstream out_points("../Data/kmeans_result.txt");
    for (const auto& p : points) {
        for (double coord : p.coords) out_points << coord << "
";
        out_points << p.cluster_id << "\n";
    }
    out_points.close();

    std::ofstream out_centers("../Data/kmeans_centers.txt");
    for (int i = 0; i < centers.size(); ++i) {
        for (double coord : centers[i].coords) out_centers <<
coord << " ";
        out_centers << i << "\n";
    }
}
```

```

out_centers.close();

    std::cout << "Результаты и центры сохранены в ../Data/" <<
std::endl;
}
std::vector<Point> kmeans(std::vector<Point>& points) {
    std::array<Point, K> centers;

    std::uniform_int_distribution<> dis(0, points.size() - 1);
    std::mt19937 gen(std::random_device {}());

    for (int i = 0; i < K; ++i) {
        centers[i] = points[dis(gen)];
        centers[i].cluster_id = i;
    }

    for (int iter = 0; iter < 100; ++iter) {
        std::for_each(std::execution::par_unseq, points.begin(),
points.end(), [&](Point& p) {
                                double    min_dist    =
std::numeric_limits<double>::max();
                                int best_id = 0;
                                for (int i = 0; i < K; ++i) {
                                    double d = distance(p, centers[i]);
                                    if (d < min_dist) {
                                        min_dist = d;
                                        best_id = i;
                                    }
                                }
                                p.cluster_id = best_id;
                            });

        std::vector<std::array<double, 9>> new_sums(K, { 0.0 });
        std::vector<size_t> counts(K, 0);

        for (const auto& p : points) {
            counts[p.cluster_id]++;
            for (size_t i = 0; i < 9; ++i) {
new_sums[p.cluster_id][i] += p.coords[i];
            }
        }

        bool changed = false;
        for (int i = 0; i < K; ++i) {
            if (counts[i] > 0) {
                for (size_t j = 0; j < 9; ++j) {
                    double new_val = new_sums[i][j] / counts[i];
                    if (std::abs(centers[i].coords[j] - new_val)
> 1e-4)
                        changed = true;
                    centers[i].coords[j] = new_val;
                }
            }
        }
    }
}

```

```

        }
    }

    if (iter % 10 == 0) {
        std::println("Итерация: {}", iter);
        std::for_each(centers.begin(), centers.end(), [&]
(Point& p) {
            std::println("Центроид {} координаты: {}",
p.cluster_id, p.coords);
        });
    }

    if (!changed) {
        if (iter % 10 != 0) {
            std::println("Итерация: {}", iter);
            std::for_each(centers.begin(), centers.end(), [&]
(Point& p) {
                std::println("Центроид {} координаты: {}",
p.cluster_id, p.coords);
            });
        }
        break;
    }
}
return { centers.begin(), centers.end() };
}

int main() {
    try {
        std::vector<Point> points = load_pointers("../Data");
        std::println("Классификация начала");
        auto start = std::chrono::high_resolution_clock::now();
        std::vector<Point> centers = kmeans(points);
        auto end = std::chrono::high_resolution_clock::now();

        std::cout << std::format(
            "Классификация завершена за {} сек\n",
            std::chrono::duration_cast<std::chrono::seconds>(end
- start).count());
        save_all_results(points, centers);
    } catch (const std::exception& e) {
        std::cerr << "Ошибка: " << e.what() << std::endl;
        return 1;
    } catch (...) {
        std::cerr << "Неизвестная ошибка" << std::endl;
        return 1;
    }

    return 0;
}

```

```

^ Projects/RV/LABA_1/build on P master IS 23
) /_main
Классификация начала
Итерация: 0
Центроид 0 координаты: [597380.7730010086, 901816.0204581473, 856829.5416255446, 301110.2994879421, 649980.5584209769, 228207.7089976502, 507141.7151419104, 211047.16351700007, 717160.891802334]
Центроид 1 координаты: [572680.4969496696, 522523.24523575197, 621104.9311085244, 682161.3417732844, 716740.6169813809, 712766.1698614657, 417951.29356861394, 622354.3778294713, 519409.7076406907]
Центроид 2 координаты: [778345.4149274716, 728243.9446739131, 709679.6169476384, 842277.74398801, 758834.9735960145, 743140.6391963116, 799166.4316213188, 726282.2467885376, 796272.4482707509]
Центроид 3 координаты: [599566.1381188119, 555012.4712399811, 264867.7889033476, 563110.493291372, 597626.8018372466, 653590.2463460632, 297099.34523856704, 652748.86562942, 462057.48161244695]
Центроид 4 координаты: [43993.23322634028, 23405.0030814738, 211757.6492526938, 168452.0493569697, 109360.5796663198, 183694.7849381356, 98869.292214112, 188063.1609315259, 122654.70188529698]
Центроид 5 координаты: [636720.9747191011, 533623.0348064919, 226098.57518137948, 573422.8248626717, 562423.9912609238, 251182.8451353805, 317954.14327513665, 519847.3309612984, 421647.2374694132]
Центроид 6 координаты: [530790.6211982659, 601864.3346820809, 240512.0581445087, 586604.8483526012, 621373.8893063584, 607497.6590780347, 704262.7563583815, 537297.4910404624, 370516.46395953745]
Центроид 7 координаты: [321347.5790618482, 531893.1765913757, 949953.4848049282, 376458.769538893, 611998.8868583163, 637773.8533675566, 409682.34429158113, 930536.1776180698, 628324.8012320328]
Центроид 8 координаты: [771282.9274177846, 346385.8095409182, 568340.9408183632, 1012417.1666666666, 660754.9092814372, 578826.5409181638, 1016798.1596806387, 563002.4908199681, 847281.5698602794]
Центроид 9 координаты: [31456.59248331076, 13170.716065241106, 147066.3179320711, 175009.6785086863, 109515.65865703691, 176140.39820944768, 113676.72566855342, 195954.64708178802, 17344.522316793016]
Итерация: 10
Центроид 0 координаты: [609148.7556049125, 901487.5142409107, 859866.6268017716, 325165.5446635337, 506285.3546149464, 227432.1437418343, 495206.80188089896, 255707.0667456884, 713478.079596823]
Центроид 1 координаты: [572513.9310812492, 525811.2252092351, 694351.87011408274, 624130.8108943895, 706016.2223008015, 733220.9250211536, 422191.10297595453, 583721.4783740803, 461202.2602545969]
Центроид 2 координаты: [790820.3111252642, 735705.2297491099, 766799.1094755979, 804264.7626105004, 791001.701051242, 832985.7427697109, 800113.3001122444, 708124.6270353146, 708926.6436467450]
Центроид 3 координаты: [780158.8093632050, 671023.6223970037, 200925.4711547735, 707025.0493445693, 653573.463295801, 283031.4014763858, 275812.34249561703, 750800.0310861423, 579118.7408008765]
Центроид 4 координаты: [45587.35464270843, 26809.85469407149, 193161.19277070076, 169016.3021650651, 109231.43084272406, 165806.51853740166, 90874.58041992657, 180787.0797152558, 105081.01133145516]
Центроид 5 координаты: [509020.977374091, 439979.4073959944, 280220.53504207876, 421353.78106083104, 491824.8663434173, 256731.55264045537, 223031.8748962792, 349525.91020830294, 331520.9095109097]
Центроид 6 координаты: [688996.348100126, 634211.7334773996, 182739.49760904026, 683513.8977129479, 634612.9570959247, 422846.8573212492, 731610.7657122277, 630661.1869432559, 438327.0809025752]
Центроид 7 координаты: [326543.8513202421, 507521.1146653543, 896283.125984252, 451665.4251230316, 607301.9247047245, 669599.8548084514, 406315.046710958, 931685.7427821523, 622796.1445209974]
Центроид 8 координаты: [758384.8034913895, 547921.7074970515, 594936.6534677991, 941634.1707949989, 677790.877188016, 572423.1950566169, 799745.4125973107, 587216.9430408116, 839288.2677518283]
Центроид 9 координаты: [25940.045234099056, 6392.390353861669, 140791.6117345583, 176438.55649170394, 103944.66725370093, 192370.69617177136, 117183.4524757529, 197626.657987698, 2501.4729193290023]
Итерация: 20
Центроид 0 координаты: [609873.3103923098, 901485.1024941544, 859081.621979735, 327946.00917930645, 650230.1621200312, 227547.12331774484, 493910.7387238764, 257941.1379542087, 713247.6408157963]
Центроид 1 координаты: [565192.1190607883, 522588.1148960196, 574819.4128144109, 661919.9048469666, 724776.6059094758, 734770.7119695117, 420689.5522762775, 618448.1768137763, 500318.9233932437]
Центроид 2 координаты: [790375.5866345575, 786640.8425045153, 766454.8134196267, 804478.6697170379, 792203.4265502709, 833425.5788681518, 801444.781577363, 780957.5526189043, 787468.803552077]
Центроид 3 координаты: [792993.288585209, 675368.6015434084, 161621.23509509666, 850861.3882636656, 609249.5062700965, 198437.596191126, 407932.0916708066, 793199.0667202573, 585837.4914581995]
Центроид 4 координаты: [45746.25636465532, 27403.26615335116, 198965.65802827207, 169048.50524006025, 109144.38608335354, 163317.13327321483, 99007.03526687795, 188804.3962953936, 104812.93201154057]
Центроид 5 координаты: [507849.7260180995, 428589.668371041, 284192.234269706, 412039.99445701356, 486502.8838235294, 238552.01345022625, 214779.4807569004, 363135.5307692308, 330025.93463687785]
Центроид 6 координаты: [330471.6495318, 591405.1837337708, 222770.4191977168, 549042.735109351, 608197.2509315175, 605410.0700780062, 489084.12229409346, 525204.7125786971, 375333.1229474406]
Центроид 7 координаты: [312761.45515019965, 504587.221680077, 906813.1971633792, 437400.852090387, 608087.4830346268, 649630.3489877171, 407741.68742832775, 934091.3805046111, 623337.8035046781]
Центроид 8 координаты: [756805.4304442779, 540332.782083776, 580809.026979351, 941010.156460177, 677264.4225132742, 575535.0468032154, 811020.346318584, 508026.823400826, 835003.8773569321]
Центроид 9 координаты: [25666.869343392384, 5741.38343352066, 140892.43745855775, 176444.72979214782, 104007.82419810117, 195218.61985337442, 117242.55930202716, 197656.89838337182, 2998.7249861124015]
Итерация: 30
Центроид 0 координаты: [609873.3103923098, 901485.1024941544, 859081.621979735, 327946.00917930645, 650230.1621200312, 227547.12331774484, 493910.7387238764, 257941.1379542087, 713247.6408157963]
Центроид 1 координаты: [565249.4347251598, 522673.72986450884, 575143.2763455024, 662216.0478923598, 724778.9093902898, 734612.8876646594, 420886.9367896136, 618420.8867143395, 500256.23013737303]
Центроид 2 координаты: [792038.921187308, 786643.6442290324, 766466.2344451803, 804492.3125413932, 792201.494378522, 833439.2558853633, 801515.7506171353, 780966.870311277, 787464.3101932687]
Центроид 3 координаты: [792993.3992609255, 675269.6572300771, 161651.38418886586, 850823.6703008433, 609158.1547236504, 198375.50695835537, 468061.7982547716, 793229.2671915167, 585794.303963609]
Центроид 4 координаты: [46145.79533926001, 27388.50505568226, 199050.06492401968, 169027.86519607843, 109166.94039215674, 162952.93578921573, 99777.3190441177, 188771.0443627451, 104998.58009756151]
Центроид 5 координаты: [507098.27948395937, 426794.3770978422, 284833.6112506223, 410677.46694828174, 485628.804297294, 236516.47686950566, 215618.95840057537, 361911.4553031168, 329467.5175465236]
Центроид 6 координаты: [630042.0485260168, 591815.4980834815, 222708.16492986022, 550000.4857359594, 607957.4243964421, 604024.0990343074, 483348.83450203313, 525136.5965692503, 375726.49325285904]
Центроид 7 координаты: [312707.4377498168, 506573.796375526, 906606.6730468071, 437184.3367235079, 608485.3462676179, 669623.4761440752, 407732.81527753606, 934105.8783713242, 623334.2545676084]
Центроид 8 координаты: [756846.1052569404, 548108.5083473127, 588260.4180330774, 941067.5823919098, 677379.0680212639, 575817.1120342587, 810869.5838157118, 587756.5853514471, 835381.6018428825]
Центроид 9 координаты: [25568.8353851023, 5883.844806232146, 141145.3211163264, 176422.81607142856, 104014.48729591841, 195410.50465586738, 117275.8565306122, 197639.6709186734, 3411.2240701224564]
Классификация завершена за 5 сек
Результаты и центры сохранены в ../Data/

```

Рисунок 1 – результат кластеризации

Вывод: В ходе лабораторной работы была успешно реализована и исследована последовательная реализация алгоритма кластеризации k-means, выполняемая на центральном процессоре. Практическая реализация позволила детально изучить архитектуру алгоритма, его базовые этапы и особенности обработки многомерных данных.

По итогам исследования были получены следующие результаты и выводы:

- Алгоритм k-means представляет собой итеративный процесс, состоящий из чередующихся шагов назначения точек кластерам и пересчета положения центроидов. Оба этапа характеризуются значительной вычислительной трудоемкостью при работе с большим количеством точек и кластеров.
- Анализ производительности CPU-реализации показал, что наибольшие временные затраты приходится на вычисление расстояний между точками и центроидами, что соответствует теоретическим оценкам вычислительной сложности алгоритма.
- При обработке массива из 80 000 точек в 9-мерном пространстве алгоритм корректно выполнил заданное количество итераций, что подтвердило правильность реализации и её устойчивость в работе с объемными данными.
- Применение стандартных контейнеров и базовых структур данных языка C++ обеспечило читаемость и надежность кода, хотя и не позволило достичь предельно возможной производительности.
- Полученные результаты указывают на перспективность применения параллельных вычислений (с использованием технологий OpenMP, CUDA или SIMD-оптимизаций), оптимизации схем доступа к памяти, а также более

совершенных методов инициализации центроидов (например, k-means++) для ускорения работы алгоритма на крупномасштабных данных.

В целом, выполнение лабораторной работы способствовало углубленному освоению методов кластерного анализа, развитию навыков работы с многомерными данными и формированию представления о вычислительных затратах классических алгоритмов машинного обучения.