

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования «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 K-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";
    }
}
```

Продолжение листинга 1

```
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;
            }
        }
    }
}
```

Продолжение листинга 1

```
        }

    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;
}
```

```

took 5 s

) /k_means
Классификация начата
Итерации: 8
Центроид 0 координаты: [597388.7730010086, 981816.8204581473, 856829.5412658466, 381118.2994897421, 649988.5584289769, 228287.7809876502, 587141.7151619184, 211087.16351780887, 717168.891802334]
Центроид 1 координаты: [572669.4969696694, 522523.24523575197, 621186.9311085264, 682161.3417732844, 716740.6169813609, 712766.169614657, 417951.29356861394, 622354.3778294713, 519469.7876466970]
Центроид 2 координаты: [778345.4149374176, 728243.9646739131, 709679.6169676384, 842277.7435988801, 758834.9735968145, 743148.6391963116, 799165.4316123188, 726282.2467885376, 796272.4482707509]
Центроид 3 координаты: [595666.1581188119, 555012.4712359981, 264867.7889833476, 563116.493291372, 597626.8918372466, 653598.2463460632, 297899.34523856704, 652748.86562942, 462807.4811244695]
Центроид 4 координаты: [459935.23522634028, 23465.8038147438, 211757.6492526958, 168452.04935669967, 189368.579666351988, 183694.7849391356, 98869.292214112, 188065.1699315259, 122654.78188529698]
Центроид 5 координаты: [656720.9747191011, 533623.0348904919, 226698.7518157948, 573422.82462628717, 562423.912669238, 251182.8451353889, 317954.1432751365, 519847.3309612984, 421647.2374694132]
Центроид 6 координаты: [530798.6211982649, 601804.334828889, 240512.0581450887, 586604.8468326012, 621373.8839635854, 607497.6598780347, 704262.7536583815, 537297.4918464624, 370516.4039595745]
Центроид 7 координаты: [321537.5979618482, 531893.1765913757, 949953.484884282, 376458.7695388093, 611998.8848581313, 637773.8535765566, 409682.34429158113, 938536.1776180698, 628524.8812328328]
Центроид 8 координаты: [346385.809546182, 568346.048183632, 1612417.1666666666, 660754.992814372, 578826.549181638, 101679.1596866387, 5633082.4980196304, 847281.708802794]
Центроид 9 координаты: [31456.597248331076, 13170.716065241186, 147606.3179320711, 175069.785886863, 105115.65865783691, 176140.39820944768, 113676.72566855342, 195954.64708178802, 17344.522316793016]
Итерации: 10
Центроид 0 координаты: [696148.75556949125, 981457.5142409197, 859664.6268617716, 325165.54640635357, 650255.3564149464, 227432.1437418343, 495286.0018808989, 255707.0667456884, 713478.9759682823]
Центроид 1 координаты: [572513.9518512492, 525811.2252082351, 443551.3812488274, 624130.8108943895, 706616.22230808105, 733220.9258211536, 422191.10297595545, 583721.1783749863, 461202.2602545699]
Центроид 2 координаты: [789280.3131525262, 785785.2297419899, 766799.189475579, 804264.7625150504, 791601.2786971242, 832985.7827691146, 64.6345767548]
Центроид 3 координаты: [780185.6983632959, 671925.62239778037, 280925.47154477535, 797625.0693445693, 653557.4632958801, 278512.7848803703, 575882.031984627, 579118.74088765]
Центроид 4 координаты: [45587.35842726843, 26889.85464967149, 198361.1927708076, 169816.3821656951, 198231.6014992744, 165886.51853748164, 99874.58414992657, 188787.8797518258, 105814.81133345536]
Центроид 5 координаты: [509820.9773374091, 439979.4073995994, 280220.53584207876, 421353.78106883104, 491894.863434173, 256731.55264045557, 223031.8748962792, 369525.91029830294, 331520.985109097]
Центроид 6 координаты: [688996.348180126, 634211.7334773996, 182739.47968094026, 663513.8977129479, 634612.5667059247, 422831.6757122492, 731610.7657122492, 630661.8766432559, 438327.0869025752]
Центроид 7 координаты: [326543.8515202421, 567521.1146655345, 896283.1259842552, 451665.4251230316, 1697301.9247047425, 669599.85488484514, 406315.064716958, 931685.742781525, 122766.144529974]
Центроид 8 координаты: [758384.8934913895, 547921.7074970515, 494134.1707949988, 677799.877188016, 572423.1950566169, 799745.4125973107, 587210.9430406111, 839289.2677518285]
Центроид 9 координаты: [256904.04523499056, 6392.39035386169, 140701.61745583, 103944.66725570963, 192370.69617177136, 17183.67424757592, 2501.2773592003]
Итерации: 20
Центроид 0 координаты: [690873.3103923098, 981485.1024941544, 859801.621979735, 327946.00917930645, 650230.16212008312, 227547.12331774484, 493910.7387238764, 257941.1379542087, 713247.6408157963]
Центроид 1 координаты: [585192.1190607983, 522588.4252082351, 442199.4128144109, 661919.9048649966, 734770.7119695117, 500318.233932437, 608198.5522762775, 1781448.1768137763, 500318.233932437]
Центроид 2 координаты: [790375.5866345578, 786640.8425045153, 766454.1834196267, 8084478.6697107379, 792203.4265502709, 833425.7868801518, 801444.781577363, 780957.5526189043, 787464.8035520277]
Центроид 3 координаты: [798293.2388585209, 675368.6015434084, 161621.2350950966, 580861.5627709065, 669269.508270579, 198437.59161120, 747932.0916780864, 73199.0667202573, 585837.4581499195]
Центроид 4 координаты: [457494.25636465532, 27403.6215335156, 266293.55802827207, 198975.65802827207, 198975.65802827207, 109144.3808333554, 261317.1332732483, 99874.3962953936, 104812.93291545057]
Центроид 5 координаты: [509820.9773374091, 439979.4073995994, 280220.53584207876, 421353.78106883104, 491894.863434173, 256731.55264045557, 223031.8748962792, 369525.91029830294, 331520.985109097]
Центроид 6 координаты: [630471.1675918095, 591405.1837337788, 222778.61917871168, 591897.2509315175, 122749.870758062, 487684.12229487346, 525204.7125786971, 375333.1229474771]
Центроид 7 координаты: [630471.1675918095, 591405.1837337788, 222778.61917871168, 591897.2509315175, 122749.870758062, 487684.12229487346, 525204.7125786971, 375333.1229474771]
Центроид 8 координаты: [756805.4304424779, 584332.782083776, 588091.026979931, 941010.1564060177, 677264.4223132742, 575535.884993154, 811020.3489318584, 588026.823480802, 835003.8773569321]
Центроид 9 координаты: [25666.86934339384, 5741.38343552866, 140892.3745855775, 176444.72979214782, 160407.3241981017, 192518.61985337442, 117242.559330282178, 197656.8983837182, 2998.742986112405]
Итерации: 30
Центроид 0 координаты: [690873.3103923098, 981485.1024941544, 859801.621979735, 327946.00917930645, 650230.16212008312, 227547.12331774484, 493910.7387238764, 257941.1379542087, 713247.6408157963]
Центроид 1 координаты: [565249.4347251598, 522588.729843550884, 575143.2673455024, 662216.8776646594, 420886.4265502709, 618420.8867143395, 500256.2307373083]
Центроид 2 координаты: [170388.921187308, 786643.6442290324, 764646.2344540183, 804494.3125843932, 792201.4947380522, 801515.7506173513, 780966.870311277, 787464.3103932667]
Центроид 3 координаты: [172913.399266925, 675265.6572300771, 161611.3841886586, 580823.6703084833, 669195.15472365804, 198735.50695835537, 468691.3876923088, 363135.53076923088, 330326.93463687785]
Центроид 4 координаты: [46145.79533926001, 27388.50550868226, 199058.06494201968, 101166.94039215674, 169027.93578921573, 99777.3190441177, 188771.04432627451, 104994.5808675151]
Центроид 5 координаты: [508098.2794839537, 426794.37709798422, 284833.6112506223, 410677.4646842924, 485628.804247924, 236516.59484405737, 329467.51759465236]
Центроид 6 координаты: [630042.0485260482, 591815.498983415, 222708.4619298602, 550900.4857359594, 687957.4243946421, 686024.8099343074, 483348.8345020331, 525136.5965692503, 375726.49325285984]
Центроид 7 координаты: [312707.4377498168, 506573.7969375326, 986606.67380488071, 437184.3362350784, 606845.3462671999, 669623.4761467052, 407732.01527753668, 934105.787313242, 623337.8035496781]
Центроид 8 координаты: [756846.88538551023, 5883.844806232146, 161145.3211163264, 176422.81697142856, 184014.48729591841, 195410.50465586738, 112725.8565306122, 197639.67891836734, 3411.2240781224564]
Центроид 9 координаты: [25368.8538551023, 5883.844806232146, 161145.3211163264, 176422.81697142856, 184014.48729591841, 195410.50465586738, 112725.8565306122, 197639.67891836734]
Классификация завершена в 3 сек
Результаты и центры сохранены в ..../Data/

```

Рисунок 1 – результат кластеризации

Вывод: В ходе лабораторной работы была успешно реализована и исследована последовательная версия 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 приходятся 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.