

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

«Метод Монте Карло»

Выполнил студент группы КС-30 Колесников Артем Максимович

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/AMKolesnikov_30_ALG

Приняли: аспирант кафедры ИКТ Пысин Максим Дмитриевич
аспирант кафедры ИКТ Краснов Дмитрий Олегович

Дата сдачи: 16.05.2022

**Москва
2022**

Содержание

Описание задачи.....	3
Описание структуры	3
Выполнение задачи	4
Заключение	9

Описание задачи

В рамках лабораторной работы необходимо решить приведенную ниже задачу с помощью метода Монте-Карло.

Пусть 2 игрока А и В играют в следующую игру: у игроков есть монетка, где 0 это орел, а 1 — это решка, каждый игрок выбирает комбинацию из 3 цифр 0/1(например, 001), затем подбрасывается монетка и результат записывается в длинную строку, побеждает тот чья комбинация будет на конце итоговой строки. Например: А - 001, В - 100, R - 01010101010100, победил В. Необходимо смоделировать игру двух игроков и построить таблицу вероятности выигрыша одной из комбинаций, так что бы столбцы соответствовали игроку А, а строки игроку В, а на пересечении была бы вероятность побед игрока А над игроком В, при выбранных ими комбинациях. Так же по результатам всех попыток определить суммарный средний шанс выигрыша игрока А и игрока В вне зависимости от выбранных комбинаций. Проанализировать полученные результаты

Описание структуры

Метод Монте-Карло (методы Монте-Карло, ММК) — общее название группы численных методов, основанных на получении большого числа реализаций стохастического (случайного) процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи.

Метод Монте-Карло является способом оценки влияния неопределенности оценки параметров системы в широком диапазоне ситуаций. Метод обычно используют для оценки диапазона изменения результатов и относительной частоты значений в этом диапазоне для количественных величин, таких как стоимость, продолжительность, производительность, спрос и др. Моделирование методом Монте-Карло может быть использовано для двух различных целей:

- трансформирование неопределенности для обычных аналитических моделей;
- расчета вероятностей, если аналитические методы не могут быть использованы.

Метод Монте-Карло может быть применен для оценки неопределенности финансовых прогнозов, результатов инвестиционных проектов, при прогнозировании стоимости и графика выполнения проекта, нарушений бизнес-процесса и замены персонала. Данный метод применяют в ситуациях, когда результаты не могут быть получены аналитическими методами или существует высокая неопределенность входных или выходных данных.

Процесс включает следующие этапы:

1. Определение модели или алгоритма, которые наиболее точно описывают поведение исследуемой системы.
2. Многократное применение модели с использованием генератора случайных чисел для получения выходных данных модели (моделирование системы). При необходимости моделируют воздействие неопределенности. Модель записывают в форме уравнения, выражающего соотношение между входными и выходными параметрами. Значения, отобранные в качестве входных данных, получают исходя из соответствующих распределений вероятностей, характеризующих неопределенности данных.
3. С помощью компьютера многократно используют модель (часто до 10000 раз) с

различными входными данными и получают выходные данные. Они могут быть обработаны с помощью статистических методов для получения оценок среднего, стандартного отклонения, доверительных интервалов.

Преимущества:

- Метод может быть адаптирован к любому распределению входных данных, включая эмпирические распределения, построенные на основе наблюдений за соответствующими системами.
- Модели относительно просты для работы и могут быть при необходимости расширены.
- Метод позволяет учесть любые воздействия и взаимосвязи, включая такие тонкие как условные зависимости.
- Для идентификации сильных и слабых влияний может быть применен анализ чувствительности.
- Модели являются понятными, а взаимосвязь между входами и выходами - прозрачной.
- Метод допускает применение эффективных моделей исследования многокомпонентных систем, таких как сеть Петри.
- Метод позволяет достичь требуемой точности результатов.
- Программное обеспечение метода доступно и относительно недорого.

Недостатки:

- Точность решений зависит от количества итераций, которые могут быть выполнены (этот недостаток становится менее значимым с увеличением быстродействия компьютера).
- Метод предполагает, что неопределенность данных можно описать известным распределением.
- Большие и сложные модели могут представлять трудности для специалистов по моделированию и затруднять вовлечение заинтересованных сторон.
- Метод не может адекватно моделировать события с очень высокой или очень низкой вероятностью появления, что ограничивает его применение при анализе риска.

Выполнение задачи

Я реализовал данную задачу на языке C++. Моя программа состоит из 3 функций: «main», «generateToss», «getLastNums».

Функция «generateToss» ~ имитирует бросок монеты N количества раз и возвращает эту последовательность «орла/решки» (0/1) в виде вектора:

```
vector<int> generateToss(int num_toss) {  
    vector<int> toss;  
  
    for (int j = 0; j < num_toss; j++)  
        toss.push_back(numGenerator(0, 1));  
  
    return toss;  
}
```

Функция «getLastNums» ~ возвращает в виде строки последние N цифр переданной последовательности:

```

string getLastNums(vector<int> toss, int num_last) {

    string str;

    for (int i = num_last; i > 0 ; i--)
        str += to_string(toss[toss.size() - i]);

    return str;
}

```

В функции «main» находится сама имитация игры человека А с человеком В ~ NUM_GAMES = { 10, 100, 1000, 10000, 100000 } число раз, для всевозможных комбинаций. После каждой игры рассчитывается вероятность победы выбранной комбинации человека А над каждой комбинацией человека В и суммарный средний шанс выигрыша каждого игрока вне зависимости от выбранных комбинаций, полученные данные выводятся в файл «Results.txt»:

```

int main()
{
    srand(time(0));
    setlocale(LC_ALL, "Rus");

    vector<string> A = { "000", "001", "010", "011", "100", "101", "110", "111" };
    vector<string> B = { "000", "001", "010", "011", "100", "101", "110", "111" };

    vector<vector<double>> win_probability_A(A.size(), vector<double>(B.size()));

    vector<int> NUM_GAMES = { 10, 100, 1000, 10000, 100000 };
    int NUM_LAST = 3;
    int NUM_TOSS = 15;

    ofstream rout("Results.txt");

    if (rout.is_open()) {
        for (int n = 0; n < NUM_GAMES.size(); n++)
        {
            cout << "Тест " << n + 1 << endl << endl;

            for (int i = 0; i < A.size(); i++)
            {
                for (int j = 0; j < B.size(); j++)
                {
                    if (A[i] == B[j])
                        continue;

                    vector<int> wins_A_B(NUM_GAMES[n]);

                    for (int k = 0; k < NUM_GAMES[n]; k++)
                    {
                        vector<int> toss = generateToss(NUM_TOSS);
                        string last_num = getLastNums(toss, NUM_LAST);

                        bool flag_A = A[i] == last_num;
                        bool flag_B = B[j] == last_num;

                        if (flag_A)
                            wins_A_B[k] = 1;

                        if (!flag_A && !flag_B) {
                            k--;
                            continue;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }

    double probability = (double)accumulate(wins_A_B.begin(),
wins_A_B.end(), 0) / NUM_GAMES[n];
    win_probability_A[i][j] = probability;
    }
}

rout << "Тест " << n + 1 << endl;
rout << "Число игр на каждую комбинацию: " << NUM_GAMES[n] << endl <<
endl;

rout << setw(10) << "A\B" << "\t";

for (int i = 0; i < B.size(); i++)
{
    rout << setw(10) << B[i] << "\t";
}
rout << "\n";

double sum_win_probability_A = 0;
double avg_win_probability_A = 0;
int count_probabilities = 0;

for (int i = 0; i < A.size(); i++)
{
    rout << setw(10) << A[i] << "\t";
    for (int j = 0; j < B.size(); j++) {
        rout << setw(10) << win_probability_A[i][j] << "\t";
        sum_win_probability_A += win_probability_A[i][j];
        if (win_probability_A[i][j] != 0)
            count_probabilities++;
    }

    rout << "\n";
}

avg_win_probability_A = sum_win_probability_A / count_probabilities;

rout << "\nСуммарный средний шанс выигрыша игрока А: " <<
avg_win_probability_A << endl;
rout << "Суммарный средний шанс выигрыша игрока В: " << 1 -
avg_win_probability_A << endl << endl << endl;
}

rout.close();
}
}

```

Результат работы программы:

Тест 1

Число игр на каждую комбинацию: 10

A\B	000	001	010	011	100	101	110	111
000	0	0.7	0.4	0.8	0.5	0.7	0.3	0.7
001	0.5	0	0.1	0.4	0.6	0.3	0.5	0.4
010	0.3	0.4	0	0.4	0.2	0.7	0.5	0.4
011	0.7	0.4	0.3	0	0.3	0.8	0.2	0.5
100	0.7	0.2	0.4	0.6	0	0	0.4	0.4
101	0.8	0.5	0.5	0.3	0.3	0	0.3	0.8
110	0.5	0.3	0.5	0.5	0.3	0.5	0	0.6
111	0.5	0.3	0.4	0.7	0.6	0.5	0.6	0

Суммарный средний шанс выигрыша игрока A: 0.472727

Суммарный средний шанс выигрыша игрока B: 0.527273

Тест 2

Число игр на каждую комбинацию: 100

A\B	000	001	010	011	100	101	110	111
000	0	0.45	0.52	0.54	0.59	0.46	0.53	0.57
001	0.49	0	0.51	0.5	0.43	0.46	0.52	0.55
010	0.5	0.44	0	0.51	0.52	0.52	0.48	0.58
011	0.54	0.47	0.53	0	0.51	0.46	0.52	0.47
100	0.56	0.61	0.49	0.5	0	0.53	0.52	0.47
101	0.47	0.52	0.48	0.51	0.46	0	0.58	0.55
110	0.48	0.49	0.47	0.51	0.52	0.52	0	0.49
111	0.5	0.52	0.53	0.59	0.49	0.49	0.43	0

Суммарный средний шанс выигрыша игрока A: 0.508036

Суммарный средний шанс выигрыша игрока B: 0.491964

Тест 3

Число игр на каждую комбинацию: 1000

A\B	000	001	010	011	100	101	110	111
000	0	0.526	0.524	0.478	0.491	0.529	0.507	0.477
001	0.473	0	0.524	0.486	0.533	0.483	0.491	0.499
010	0.512	0.483	0	0.501	0.505	0.489	0.475	0.489
011	0.507	0.487	0.522	0	0.518	0.53	0.487	0.498
100	0.518	0.496	0.484	0.49	0	0.5	0.477	0.502
101	0.485	0.499	0.459	0.472	0.477	0	0.46	0.506
110	0.477	0.473	0.491	0.506	0.518	0.476	0	0.511
111	0.516	0.506	0.513	0.493	0.512	0.487	0.521	0

Суммарный средний шанс выигрыша игрока A: 0.497304

Суммарный средний шанс выигрыша игрока B: 0.502696

Тест 4

Число игр на каждую комбинацию: 10000

A\B	000	001	010	011	100	101	110	111
000	0	0.4959	0.5061	0.5002	0.4968	0.5124	0.4993	0.4939
001	0.4972	0	0.4971	0.5108	0.4964	0.5027	0.5036	0.5007
010	0.4939	0.4998	0	0.4981	0.5116	0.4942	0.4988	0.5058
011	0.4978	0.4898	0.4969	0	0.5053	0.4912	0.5017	0.5062
100	0.5018	0.5021	0.4909	0.4999	0	0.5021	0.4997	0.5002
101	0.5072	0.4939	0.4962	0.5072	0.5001	0	0.4901	0.5069
110	0.5021	0.5024	0.5025	0.4994	0.5031	0.5081	0	0.4957
111	0.5013	0.5013	0.5	0.5034	0.5046	0.4949	0.5109	0

Суммарный средний шанс выигрыша игрока A: 0.500575

Суммарный средний шанс выигрыша игрока B: 0.499425

Тест 5

Число игр на каждую комбинацию: 100000

A\B	000	001	010	011	100	101	110	111
000	0	0.50012	0.49975	0.50011	0.5002	0.50014	0.49989	0.50038
001	0.49976	0	0.50039	0.49994	0.49995	0.49979	0.50018	0.49992
010	0.49995	0.49961	0	0.49997	0.49959	0.50029	0.5	0.50014
011	0.49969	0.50011	0.5001	0	0.50033	0.50033	0.49989	0.50006
100	0.5	0.49979	0.5	0.50008	0	0.49997	0.50016	0.5004
101	0.50039	0.49988	0.49967	0.50004	0.49997	0	0.50007	0.49989
110	0.50001	0.49988	0.49993	0.49999	0.5002	0.50003	0	0.49988
111	0.49989	0.50016	0.50009	0.50044	0.49979	0.4999	0.5003	0

Суммарный средний шанс выигрыша игрока A: 0.500025

Суммарный средний шанс выигрыша игрока B: 0.499975

Рисунок 1 - Результат из файла "Results.txt"

Число игр на каждую комбинацию	10	100	1000	10000	100000
Суммарный средний шанс победы А	0,472727	0,508036	0,497304	0,500575	0,500025
Суммарный средний шанс победы В	0,527273	0,491964	0,502696	0,499425	0,499975
Разница	0,054546	0,016072	0,005392	0,00115	0,00005

Таблица 1 - Суммарный средний шанс победы каждого игрока за N игр

Суммарный средний шанс победы

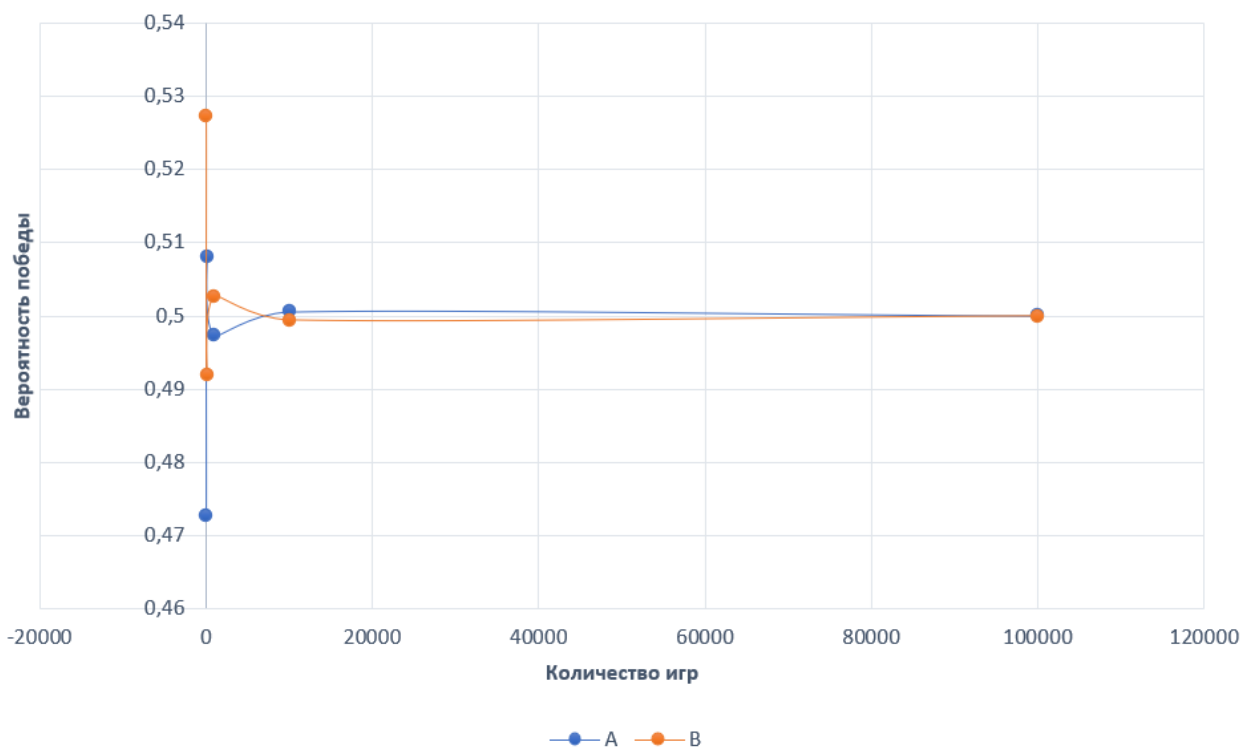


Рисунок 2 – График суммарного среднего шанса победы каждого игрока за N игр.

Заключение

В этой лабораторной работе мне удалось решить задачу при помощи метода Монте-Карло. Анализируя полученные результаты наглядно видно, как с увеличением числа итераций (в нашем случае ~ числа игр) – возрастает точность решения поставленной задачи. Если при 10 играх на каждую комбинацию разница между суммарным средним шансом выигрыша каждого из игроков составляет 0.054546, то при 100000 игр эта разница уменьшается до 0.00005 и вероятность победы каждого игрока стремится приблизиться к 0.5, как и должно быть по аналитическим расчетам. Метод Монте-Карло прекрасно справляется с поставленной задачей.