Тестовое задание.

Обозначения.

Рассмотрим некоторое конечное дискретное вероятностное распределение случайного числа x (x, p) = {(x1, p1), . . . ,(xm, pm)}, Xm n=1 pn = 1, pn ≥ 0, где {x1, . . . , xm} — множество всевозможных исходов, {p1, . . . , pm} — отвечающие этим исходам вероятности. Числа Es(x) = Xm n=1 x s npn будем называть s-ыми моментами x, а числа E(x) = E1(x), D(x) = E2(x) − E 2 1 (x), соответственно, математическим ожиданием x и дисперсией x.

Задание 1.

Дано: (x, p) и (y, q), отвечающие независимым случайным числам x и y, для которых известны лишь E(x), D(x), E(y) и D(y).

Найти: для случайного числа z = x − 2y значения E(z) и D(z).

Решение:

математическое ожидание суммы случайных величин: E(x + y) = E(x) + E(y),

дисперсия суммы случайных величин: D(x + y) = D(x) + D(y), если x и y независимы,

математическое ожидание произведения случайных величин: E(aX) = aE(x), где "a" – константа,

дисперсия произведения случайных величин: D(aX) = a^2 \* D(x), где "a" – константа.

Исходя из вышеуказанных формул, рассчитывается E(z) и D(z):

E(z) = E(x – 2y) = E(x) - 2E(y)

D(z) = D(x – 2y) = D(x) + 4D(y)

Примем, что E(x) = 10, D(x)= 5, E(y) = 3, D(y) = 2,

тогда: E(z) = 10 – 2\*3 = 4;

D(z) = 5 + 4\*2 = 13.

Ответ: E(z) = 4, D(z) = 13.

Код на языке C++, который будет вычислять E(z) и D(z):

#include <iostream>

int main() {

double E\_x = 10;

double D\_x = 5;

double E\_y = 3;

double D\_y = 2;

// Расчет E(Z) и D(Z)

double E\_z = E\_x - 2 \* E\_y;

double D\_z = D\_x + 4 \* D\_y;

// Вывод результата

std::cout << "E(Z) = " << E\_z << std::endl;

std::cout << "D(Z) = " << D\_z << std::endl;

return 0; // Возвращаемое значение 0 для указания, что программа завершилась без ошибок.

}

Задание 2.

Дано: (x, p) и (y, q), отвечающие независимым случайным числам x и y, для которых известны лишь E(x), D(x), E(y) и D(y). Для некоторого действительного α ∈ [0, 1] cтроится смесь этих случайных распределений (z, r) = (zα, rα). Процесс построения этой смеси выглядит следующим образом, каждый раз для определения значения z случайным образом выбирается равномерно распределенное на отрезке [0, 1] число σ, если данное число σ ≤ α, то выбирается событие из потока, отвечающего x, если же оказалось выполненным σ > α, то выбирается событие из потока, отвечающего y.

Найти: E(z) и D(z) для случайного числа z.

Решение:

Формула полного математического ожидания:

E(z) = α \* E(x) + (1 - α) \* E(y)

Формула полной дисперсии:

D(z) = α \* (D(x) + (E(x))^2) + (1 - α) \* (D(y) + (E(y))^2) - (E(z))^2.

Примем, что α = 0,5, E(x) = 10, D(x) = 5, E(y) = 3, D(y) = 2,

тогда : E(z) = 0,5 \* 10 + 0,5 \* 3 = 6,5;

D(z) = 0,5 \* (5 + 100) + 0,5 \* (2 + 9) – 42,25 = 52,5 + 5,5 – 42,25 = 15,75.

Ответ: E(z) = 6,5, D(z) = 15,75.

Код на языке C++:

#include <iostream>

double calculateExpectedValue(double alpha, double E\_x, double E\_y) {

return alpha \* E\_x + (1 - alpha) \* E\_y;

}

double calculateVariance(double alpha, double E\_x, double D\_x, double E\_y, double D\_y, double E\_z) {

return alpha \* (D\_x + E\_x \* E\_x) + (1 - alpha) \* (D\_y + E\_y \* E\_y) - E\_z \* E\_z;

}

int main() {

// Входные данные

double alpha = 0.5; // Значение α ∈ [0, 1]

double E\_x = 10; // Математическое ожидание x

double D\_x = 5; // Дисперсия x

double E\_y = 3; // Математическое ожидание y

double D\_y = 2; // Дисперсия y

// Расчет E(z) и D(z)

double E\_z = calculateExpectedValue(alpha, E\_x, E\_y);

double D\_z = calculateVariance(alpha, E\_x, D\_x, E\_y, D\_y, E\_z);

// Вывод результатов

std::cout << "E(z) = " << E\_z << std::endl;

std::cout << "D(z) = " << D\_z << std::endl;

return 0;

}

Задание 3.

Дано:

Игра бинго. Предположим, что вы играете на этой карточке, той что 5 × 5 на рисунке 1. Среднее поле считается открытым уже до начала игры. Последовательно из кучи, в которой 75 шаров с номерами 1-75, случайным образом выбирается Рис. 1: 1 шар за шаром, вы отмечаете на своей карте выпавшие шары последовательно. Выбранные шары удаляются из кучи. Предположим, что карта считается выигравшей, если вы собрали квадрат 3 × 3 с центром в центральной позиции карточки. Предположим также, что стоимость карточки 1$, а за то, что вы собрали выигрышный комплект на s-ом шаре (s = 1, 2, . . . , 75) вам выплачивается выигрыш Ws и игра заканчивается. Ваша задача стандартными средствами MS Excel посчитать вероятности получения выигрыша Ws. И придумать таблицу выигрышей, так чтобы средний выигрыш в игре был примерно равен 0.95$±0.02$, а вероятность получения положительного выигрыша была примерно равна 20%. В вашей таблице должно быть не менее 7 различных положительных значений, но могут быть нулевые, например: W1 = · · · = W10 = 100$, W11 = · · · = W13 = 10$, W14 = · · · = W18 = 1$, W19 = · · · = W75 = 0$. Значения в таблице должны также удовлетворять естественному ограничению Ws+1 ≤ Ws при s = 1, . . . , 74. Как факт, в руках у вас окажется конечное дискретное вероятностное распределение случайного числа W. В том же MS Excel посчитайте для него E(W) и D(W).

Решение находится в файле Задание 3 (Кузнецова Д.А.).

Задание 4.

Предположим есть игра, стоимость которой равна 1. Призами являются Ws ∈ Ω = {0, 1, 2, 3, 5, 7, 10, 50, 100, 200, 500}, W1 = 0. Вероятности ps каждого события строго положительны и совокупность всех Ws дает множество Ω. Вам нужно найти такие вероятности, чтобы выполнялись E(W) = 0.95 ± 0.005, D(W) = 154.00 ± 0.5 и вероятность появления положительного выигрыша в игре (1 − p1) была бы равна 0.15; опишите ваш алгоритм поиска этих вероятностей. Далее средствами языка C + + необходимо реализовать мультипоточное моделирование данного игрового процесса, так чтобы данные, необходимые для построения гистограммы распределения и полученные от различных потоков, запущенных на различных ядрах процессора, время от времени сбрасывались в некий общий аккумулятор. Например, на каждом ядре расcчитывается 10000 итераций, затем данные сбрасываются в общий аккумулятор, каждое ядро совершает 10000 таких циклов и останавливает свою работу. Когда все такие потоки завершены данные аккумулятора сбрасываются в файл вместе с итоговым количеством испытаний. Вас интересуют относительные погрешности расчета E(W) и D(W) по аккумулируемой выборке. В исходном коде нужно использовать класс std::mt19937 для генератора случайных чисел и класс std::mutex для операций с аккумулятором.

Решение:

При решении задачи использован метод Монте-Карло:

Инициализация вероятностей ps для каждого события Ws случайными значениями, при условии что сумма всех вероятностей равна 1.

Цикл:

генерация случайного выигрыша W в соответствии с вероятностями ps и набором значений Ws;

вычисление среднего значения E(W) и дисперсии D(W) для полученного выигрыша W;

вычисление вероятности появления положительного выигрыша (1 - p1) на основе текущих вероятностей ps;

если полученные значения E(W), D(W), и (1 - p1) удовлетворяют заданным диапазонам, то цикл заканчивается;

если нет, вероятности обновляются путем регулировки их значений.

Код находится в файле test.cpp.

Результат выполнения программы:

P(W1) = 0.292, P(W2) = 0.038, P(W3) = 0.061, P(W4) = 0.098, P(W5) = 0.106, P(W6) = 0.098, P(W7) = 0.086, P(W8) = 0.032, P(W9) = 0.027, P(W10) = 0.091, P(W11) = 0.071.

Мультипоточное моделирование данного игрового процесса:

Код находится в файле test1.cpp. Для реализации мультипоточного моделирования игрового процесса на языке C++, используется мьютекс (std::mutex) для синхронизации доступа к общему вектору accumulator, где сохраняются результаты каждой итерации моделирования игры:

В этом коде создается несколько потоков, которые выполняют функцию simulateGame для моделирования игрового процесса с заданным количеством итераций. Результаты каждой итерации сохраняются в общем векторе accumulator, который синхронизируется с помощью мьютекса accumulator\_mutex. После завершения всех потоков, результаты выводятся на экран.

Задание 5. Есть игра, которую легко представить на матрице 3 × 3. Итак, есть 9 ячеек, в которых прячутся 9 предметов: AABBCCDDS. Мы последовательно открываем ячейки и при получении, либо одной из пар AA, BB, CC или DD, либо открытия ячейки с S игра заканчивается. Стоимость игры равна 1, за пары игрок получает фиксированный приз AA — 10, BB — 20, CC — 30, DD — 40, а за нахождение S — случайным образом либо 50, либо 100, либо 200. Требуется определить вероятность нахождения игроком S. И найти такое распределение для призов S, чтобы средний выигрыш в игре был равен 55.

Решение находится в файле Задание 5 (Кузнецова Д.А.).