Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа компьютерных технологий и информационных систем

Отчёт по практической работе

Дисциплина: Теория вероятностей и математическая статистика

Выполнил студент гр. 5130901/20004 ________________ Самохвалова П. А.
(подпись)

Преподаватель ________________________ Никитин К. В.
(подпись)

“____” _____________ 2024 г.

Санкт-Петербург
2024

Список задач для аналитического решения: **1.1, 2.15, 3.18, 4.12, 5.33**

№ 1.1

- Что означают события А + А и АА?

Решение: Суммой А + А множеств А является само множество А (произойдет А или А). Аналогично, произведением АА множеств А является само множество А (произойдет А и А).

№ 2.15

- Имеются $n + m$ билетов, из которых n выигрышных. Одновременно приобретаются k билетов. Определить вероятность, что среди них s выигрышных.

Решение: Общее число исходов – кол-во комбинаций вытянуть k билетов из $n + m$, значит $n = C_{n+m}^{k}$. Благоприятствующие исходы – произведение кол-ва комбинаций вытянуть s выигрышных билетов из n и кол-ва комбинаций вытянуть $k - s$ невыигрышных билетов из m, значит $m = C_n^s C_m^{k-s}$. Искомая вероятность $P(A) = \frac{C_n^s C_m^{k-s}}{C_{n+m}^{k}}$.

№ 3.18

- Два судна в тумане: одно идет вдоль пролива шириной L, а другое курсирует без остановок поперек этого пролива перпендикулярно курсу первого. Скорости движения судов соответственно равны v_1 и v_2. Второе судно подает звуковые сигналы, которые слышны на расстоянии $d < L$. Определить вероятность того, что на первом судне услышат сигналы, если пересечение курсов судов равновозможно в любом месте пролива.

Решение: пусть x – расстояние от берега до первого судна, и y – расстояние от берега до второго судна. С учетом ширины пролива, x и y могут принимать значения от 0 до L, то есть $0 \le x \le L$, $0 \le y \le L$ – областью возможных значений является квадрат площадью L^2. Для того, чтобы найти благоприятствующие значения x и y (т. е. такие расстояния от одного берега, что на первом судне услышат второе), построим вектор относительной скорости для первого относительно второго. Также вокруг второго судна строится окружность радиуса d – область, в которой слышны сигналы. Тогда на первом судне услышат сигналы, если прямая, проходящая через вектор относительной скорости, пересечет окружность. Изображение для двух ситуаций (второе судно движется вверх/вниз) приведено на рис. 1.

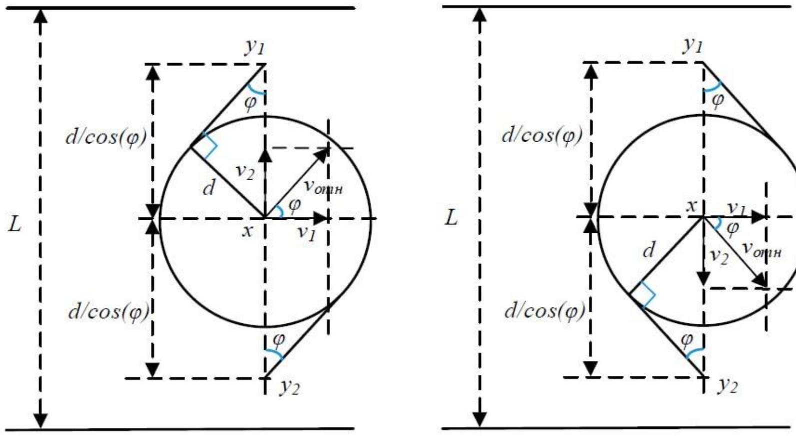

Рисунок 1.

Из рисунка следует, что прямая пересечет окружность в случае, когда $|x-y|<\frac{d}{\cos\varphi}$. Угол φ также получается между вектором относительной скорости и вектором скорости первого судна, значит можно записать, что $\tan\varphi=\frac{v_2}{v_1}$. Для того, чтобы выразить угол φ через скорости судов, возьмем формулу связи между тангенсом и косинусом: $\tan^2\varphi+1=\frac{1}{\cos^2\varphi}$, $\cos^2\varphi=\frac{1}{\tan^2\varphi+1}$, $\cos\varphi=\sqrt{\frac{1}{\tan^2\varphi+1}}$. Отсюда $\cos\varphi=\sqrt{\frac{1}{\left(\frac{v_2}{v_1}\right)^2+1}}$, значит условие можно изменить на $|x-y|<\frac{d}{\sqrt{\frac{1}{\left(\frac{v_2}{v_1}\right)^2+1}}}$, $|x-y|<\frac{d}{\frac{1}{\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}}}$, $|x-y|<d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}$.

График благоприятных значений x и y с приведен на рис. 2, соответствующая область залита синим цветом.

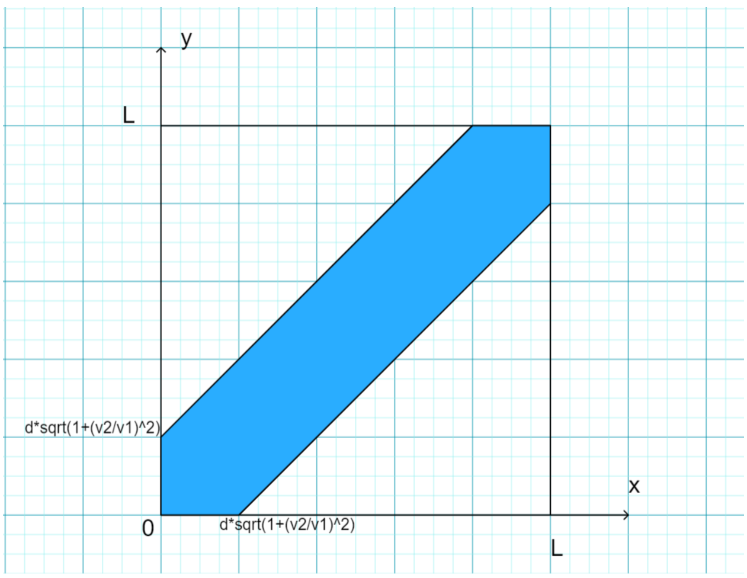

Рисунок 2.

Площадь области – $L^2-2\frac{\left(L-d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}\right)\left(L-d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}\right)}{2}=L^2-\left(L-d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}\right)^2$. Теперь можно найти искомую вероятность по формуле площадь благоприятствующей области/площадь возможных значений:

$$\frac{L^2-\left(L-d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}\right)^2}{L^2}=1-\frac{\left(L-d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}\right)^2}{L^2}$$
$$=1-\left(\frac{L-d\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}}{L}\right)^2=1-\left(1-\frac{d}{L}\sqrt{\left(\frac{v_2}{v_1}\right)^2+1}\right)^2$$

№ 4.12

- Вероятность того, что что в электрической цепи напряжение превысит номинальное значение, равна p_1. При повышенном напряжении вероятность аварии прибора-потребителя электрического тока равна p_2. Определить вероятность аварии прибора вследствие повышения напряжения.

Решение: пусть А – превышение напряжения в сети, В – авария прибора. Тогда, по условию задачи, $P(A)=p_1$, $P(B|A)=p_2$. Вероятность аварии прибора вследствие повышения напряжения – $P(BA)$. Следовательно, $P(BA)= P(A)P(B|A)=p_1p_2$.

№ 5.33

- В помещении, насчитывающем n пронумерованных мест, n лицам выдали n номерных билетов. Какова вероятность, что ровно m лиц окажутся сидящими на местах, соответствующим номерам билетов, если все места занимаются наудачу?

Решение: вероятность m лицам сесть на свои места – $P(A)= C_n^m\frac{(n-m)!}{n!}=\frac{n!}{(n-m)!m!}\frac{(n-m)!}{n!}=\frac{1}{m!}$, т. е. m человек сели на свои места и $(n-m)$ сели на оставшиеся (C_n^m) с учетом всех возможных способов для m человек занять любые места ($n(n-1)(n-2)\dots(n-m+1)=\frac{n!}{(n-m)!}$ – у первого человека выбор из n мест, у второго из $(n-1)$ и т.д.). Но $(n-m)$ людей могут занять и свои места, значит нужно вычислить вероятность $P(B)$ того, что они займут другие. Вероятность $P(B_k)$ занять каждое место одинакова, значит можно применить формулу $P(\sum_{k=1}^{n}B_k)=\sum_{k=1}^{n}(-1)^{k-1}C_n^kP(B^k)$, где $n = n - m$.

Тогда $P(\sum_{k=1}^{n-m} B_k) = \sum_{k=1}^{n-m} (-1)^{k-1} C_{n-m}^{k} P(B^k)$. $P(B^k)$ – вероятность того, что хотя бы один из k лиц занял не свое место, тогда $P(\overline{B^k})$ – вероятность того, что все k человек заняли свои места из $(n - m)$ оставшихся, $P\left(\overline{B^k}\right) = \frac{1}{n-m} * \frac{1}{n-m-1} * \frac{1}{n-m-2} \dots \frac{1}{n-m-k+1} = \frac{(n-m-k)!k!}{(n-m)!k!}$. Значит $P(B^k) = 1 - \frac{(n-m-k)!k!}{(n-m)!k!}$,

$$P\left(\sum_{k=1}^{n-m} B_k\right) = \sum_{k=1}^{n-m} (-1)^{k-1} C_{n-m}^{k} (1 - \frac{(n-m-k)!\,k!}{(n-m)!\,k!})$$
$$= \sum_{k=1}^{n-m} (-1)^{k-1} C_{n-m}^{k} \left(1 - \frac{1}{C_{n-m}^{k}\,k!}\right) = \sum_{k=1}^{n-m} (-1)^{k-1} \left(C_{n-m}^{k} - \frac{1}{k!}\right)$$
$$= \sum_{k=1}^{n-m} (-1)^{k-1} \left(C_{n-m}^{k} - \frac{1}{k!}\right) = \sum_{k=1}^{n-m} (-1)^{k-1} C_{n-m}^{k} + \sum_{k=1}^{n-m} \frac{(-1)^k}{k!}$$
$$= 1 - \sum_{k=0}^{n-m} (-1)^k 1^{n-k} C_{n-m}^{k} + \sum_{k=1}^{n-m} \frac{(-1)^k}{k!}$$
$$= 1 - (1-1)^{n-m} + \sum_{k=1}^{n-m} \frac{(-1)^k}{k!} = \sum_{k=1}^{n-m} \frac{(-1)^k}{k!}$$

Искомая вероятность является произведением вероятностей событий A и B: $P(A)\,P(B) = \frac{1}{m!} \sum_{k=1}^{n-m} \frac{(-1)^k}{k!}$.

Список задач для решения с помощью имитационного моделирования: **2.15, 3.18, 4.12, 5.33**

№ 2.15

Моделирование:

Параметры – `n` – количество выигрышных билетов, `m` – количество проигрышных билетов, `k` – количество выбранных билетов, `s` – количество выигрышных билетов, которые должны быть выбраны для благоприятствующего исхода – гененируются случайно функцией `random.randint()`. `num` – количество опытов. `good` – для подсчета благоприятствующих исходов. `tickets` – массив размера `n+m`, состоящий из `n` единиц и `m` нулей.

Перед выполнением моделирования для каждого случайного набора начальных данных считается результат по формуле, полученной для аналитического решения. Для каждого опыта `num` проводится `k` выборов, где из массива `tickets` выбирается «билет» – ноль (проигрышный) или единица (выигрышный). После выбора соответствующее число удаляется из массива. Переменная `chosen` подсчитывает количество выбранных выгрышных билетов, потом они сравниваются с `s`. В листинге 1 приведен код программы.

Листинг 1. Код программы на языке Python

```
import random
import math

n = random.randint(1, 100)
m = random.randint(1, 100)
k = random.randint(1, n + m)
s = random.randint(1, m)
while s > k or s > n:
    s = random.randint(1, m)
print("n =", n, "m =", m, "k =", k, "s =", s)
print("Аналитическое решение:", (math.comb(n, s) * math.comb(m, k - s)) /
math.comb(n + m, k))
good = 0
num = 1000000
for i in range(num):
    tickets = n * [1] + m * [0]
    chosen = 0
    for j in range(k):
        if random.choice(tickets) == 0:
            tickets.remove(0)
        else:
            tickets.remove(1)
            chosen += 1
    if chosen == s:
        good += 1
print("Результат моделирования:", good / num)
```

n = 15 m = 46 k = 10 s = 4 Аналитическое решение: 0.1417843108511472 Результат моделирования: 0.142157
n = 79 m = 71 k = 31 s = 9 Аналитическое решение: 0.002054009981772243 Результат моделирования: 0.002003
n = 1 m = 67 k = 33 s = 1 Аналитическое решение: 0.4852941176470588 Результат моделирования: 0.48467
n = 23 m = 64 k = 27 s = 11 Аналитическое решение: 0.028394047227916608 Результат моделирования: 0.028345
n = 99 m = 66 k = 6 s = 2 Аналитическое решение: 0.13675842013154726 Результат моделирования: 0.136078

Таблица 1. Результаты моделирования

В табл. 1 приведены результаты моделирования. С небольшой погрешностью, они совпадают с аналитическим решением.

№ 3.18

Моделирование:

Параметры – `L` – ширина пролива, `d` – расстояние, на котором слышны сигналы от второго судна, `v1, v2` – скорости судов – гененируются случайно функцией `random.randint()`. `num` – количество опытов. `good` – для подсчета благоприятствующих исходов.

Перед выполнением моделирования для каждого случайного набора начальных данных считается результат по формуле, полученной для аналитического решения. Для каждого опыта `num` случайно генерируются положения судов – расстояния от одного берега `x` и `y`. Если значение расстояния между `x` и `y` входит в благоприятствующую область, счетчик `good` увеличивается на единицу. В листинге 2 приведен код программы.

Листинг 2. Код программы на языке Python

```
import math
import random
```

```python
L = random.randint(1, 1000)
d = random.randint(0, L)
v1 = random.randint(1, L)
v2 = random.randint(1, L)
print("L =", L, "v1 =", v1, "v2 =", v2, "d =", d)
print("Аналитическое решение:", 1 - (1 - d / L * math.sqrt(1 + (v2 / v1) **
2)) ** 2)
good = 0
num = 1000000
for i in range(num):
    x = random.randint(0, L)
    y = random.randint(0, L)
    if abs(x - y) <= d * math.sqrt(1 + (v2 / v1) ** 2):
        good += 1
print("Результат моделирования:", good / num)
```

L = 792 v1 = 509 v2 = 593 d = 421 Аналитическое решение: 0.9661948750878667 Результат моделирования: 0.965918
L = 963 v1 = 310 v2 = 43 d = 914 Аналитическое решение: 0.9982531383768545 Результат моделирования: 0.998208
L = 99 v1 = 45 v2 = 2 d = 28 Аналитическое решение: 0.48606511705637145 Результат моделирования: 0.488542
L = 433 v1 = 207 v2 = 22 d = 156 Аналитическое решение: 0.593346584419186 Результат моделирования: 0.59124
L = 64 v1 = 45 v2 = 44 d = 18 Аналитическое решение: 0.6319793863779781 Результат моделирования: 0.630541

Таблица 2. Результаты моделирования

В табл. 2 приведены результаты моделирования. С небольшой погрешностью, они совпадают с аналитическим решением.

№ 4.12

Моделирование:

Параметры – `p1` – вероятность того, что что в электрической цепи напряжение превысит номинальное значение, `p2` – вероятность аварии прибора-потребителя электрического тока – гененируются случайно функцией `random.random()`. `num` – количество опытов. `good` – для подсчета благоприятствующих исходов.

Перед выполнением моделирования для каждого случайного набора начальных данных считается результат по формуле, полученной для аналитического решения. В каждом опыте `num` с заданной вероятностью `p1` может произойти событие A – выбрана единица; если оно происходит, с заданной вероятностью `p2` может произойти такое же событие B. Если оба раза выбрана единица, то исход считается благоприятным. В листинге 3 приведен код программы.

Листинг 3. Код программы на языке Python

```python
import random

p1 = random.random()
p2 = random.random()
print("p1 =", p1, "p2 =", p2)
print("Аналитическое решение:", p1 * p2)
good = 0
num = 1000000
for i in range(num):
    a = random.choices([1, 0], weights=[p1, 1 - p1], k=1)[0]
    if a == 1:
        b = random.choices([1, 0], weights=[p2, 1 - p2], k=1)[0]
        if b == 1:
            good += 1
print("Результат моделирования:", good / num)
```

```
p1 = 0.7616156771728008 p2 = 0.8155444377053784
Аналитическое решение: 0.6211314291874929
Результат моделирования: 0.621709
```

```
p1 = 0.36935656187941424 p2 = 0.9123265003356456
Аналитическое решение: 0.3369737794754523
Результат моделирования: 0.336497
```

```
p1 = 0.9868508584048489 p2 = 0.41413152414736
Аналитическое решение: 0.4086860500973306
Результат моделирования: 0.408174
```

```
p1 = 0.6681303872681359 p2 = 0.5071966683700468
Аналитическое решение: 0.3388735064591877
Результат моделирования: 0.338391
```

```
p1 = 0.7767764584465614 p2 = 0.961777685522691
Аналитическое решение: 0.7470862643732467
Результат моделирования: 0.746834
```

Таблица 3. Результаты моделирования

В табл. 3 приведены результаты моделирования. С небольшой погрешностью, они совпадают с аналитическим решением.

№ 5.33

Моделирование:

Параметры – `n` – количество мест, `m` – количество мест, которые должны быть правильно заняты – гененируются случайно функцией `random.random()`. `num` – количество опытов. `good` – для подсчета благоприятствующих исходов.

Перед выполнением моделирования для каждого случайного набора начальных данных считается результат по формуле, полученной для аналитического решения. Изначально имеется массив `seats` с числами от 1 до `n`. В каждом опыте `num` в массив `tickets` копируется массив `seats` и он перемешивается, таким образом имитируя занятие мест наудачу. После сравниваются пары элементов, имеющие один индекс, т. е. проверяется, соответствует ли билет месту. Если да, то счетчик `correct` увеличивается на единицу. Если число правильно занятых мест равно `m`, исход считается благоприятным. В листинге 3 приведен код программы.

Листинг 4. Код программы на языке Python

```python
import random
import math


def calc_sum(x, y):
    result = 0
    for j in range(x - y):
        result += (-1) ** j / math.factorial(j)
    return result


n = random.randint(10, 200)
m = random.randint(1, 10)
print("n =", n, "m =", m)
print("Аналитическое решение:", (1 / math.factorial(m)) * calc_sum(n, m))
num = 1000000
good = 0
seats = list(range(1, n + 1))
for i in range(num):
    correct = 0
    tickets = seats[:]
    random.shuffle(tickets)
    for k in range(n):
        if seats[k] == tickets[k]:
            correct += 1
    if correct == m:
        good += 1
print("Результат моделирования:", good / num)
```

```
n = 79 m = 2
Аналитическое решение: 0.18393972058572122
Результат моделирования: 0.183885

n = 49 m = 3
Аналитическое решение: 0.061313240195240405
Результат моделирования: 0.061059

n = 191 m = 4
Аналитическое решение: 0.015328310048810101
Результат моделирования: 0.015304

n = 182 m = 1
Аналитическое решение: 0.36787944117144245
Результат моделирования: 0.367402

n = 23 m = 5
Аналитическое решение: 0.003065662009762019
Результат моделирования: 0.003041
```

Таблица 4. Результаты моделирования

В табл. 4 приведены результаты моделирования. С небольшой погрешностью, они совпадают с аналитическим решением.