Probability Theory tasks #1-5

Vusal Aliev

1. Problems to Solve

#1.7. Для решения воспользуемся выражением: $\bar{A} = \Omega \setminus A = \{\forall x \in \Omega : x \notin A\}$. Исходя из него можно сделать вывод, что противоложное событие $\bar{A}$ происходит тогда и только тогда, когда событие A не происходит.
Для события А из задания противоположным событием $\bar{A}$ является событие, при котором ни одно из имеющихся четырех изделий не является бракованным. А для события B противоположным событием $\bar{B}$ является событие, при котором бракованных изделий среди них не более одного.

#2.24. Для вычисления вероятности из задачи необходимо сперва подсчитать общее количество способов взять m изделий из n – это равно $C_n^m = n_{\text{общ}}$.

$$\overbrace{\begin{array}{|c|c|c|c|c|}\hline n_1 & n_2 & n_3 & ... & n_k \\ \hline\end{array}}^{n}$$
$$\downarrow$$
$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline m_1 & m_2 & m_3 & ... & m_k \\ \hline\end{array}}_{m}$$

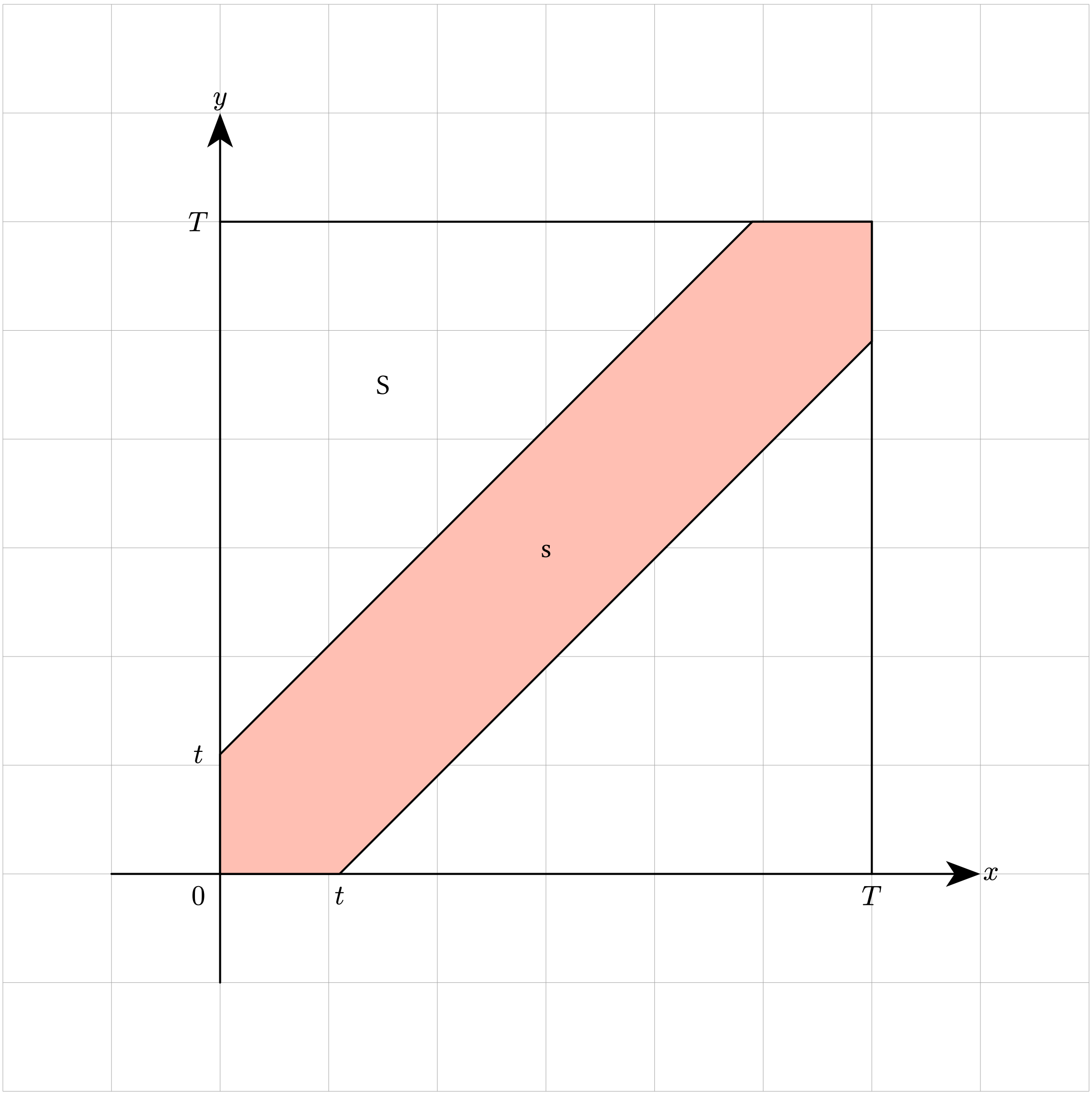
Для подсчета количества благоприятных исходов необходимо вспомнить, что если m элементов берутся из совокупности n элементов, которая объединяет k различных признаков (групп), состоящих соответственно из $j_1, j_2, ..., j_k$ элементов, то число способов выбрать любые $m_1, m_2, ..., m_k$ элементов из указанных групп, где $\sum_{j=1}^{k} m_j = m$, равно $\prod_{j=1}^{k} C_{n_j}^{m_j} = n_{\text{бл}}$.

Следовательно искомая вероятность равна $p = \frac{n_{\text{бл}}}{n_{\text{общ}}} = \frac{\prod_{j=1}^{k} C_{n_j}^{m_j}}{C_n^m}$.

#3.17. Попробуем представить задачу на координатной плоскости. Обозначим момент подхода одного лица за x, а момент подхода другого лица за y, где

$$0 \leq x \leq T \quad \text{и } 0 \leq y \leq T$$

тогда каждый случай их общего подхода можно задать парой (x, y) или же точкой на координатной плоскости. Все такие «точки», то есть исходы, образуют квадрат со стороной T.



Из всех этих исходов благоприятными являются те, которые удовлетворяют следующему неравенству:

$$\mid x - y \mid \leq t \tag{1}$$

Для вычисления геометрической вероятности осталось посчитать площади, получившихся фигур.

Площадь $s = S - 2 \cdot$ (площади равнобедренного треугольника с катетом $a = T - t$) $=$

$$= S - 2 \cdot \frac{(T-t)^2}{2}$$

Искомая вероятность:

$$p = \frac{s}{S} = \frac{S - (T-t)^2}{S} = 1 - \frac{(T-t)^2}{T^2} = 1 - \left(1 - \frac{t}{T}\right)^2$$

#4.32. Наша вероятность равна

$$P = \prod_{i=1}^{50} p_i \text{ , где } p_i \text{ — вероятность не вытащить черный шар на } i\text{-ой попытке}$$

$p_i = 1$ — вероятность вытащить черный шар на i-той попытке

$= 1 - \frac{1}{\text{число всех шаров}}$

$= 1 - \frac{1}{2i}$

Тогда искомая вероятность равна

$$\begin{aligned} P = p_1 \cdot p_2 \cdot p_3 \cdot \ldots \cdot p_{50} = \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{4}\right) \cdot \left(1 - \frac{1}{6}\right) \cdot \ldots \cdot \left(1 - \frac{1}{100}\right) = \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdot \ldots \cdot \frac{99}{100} \\ = \frac{100!}{2^{50} \cdot 50!} \cdot \frac{1}{2^{50} \cdot 50!} \\ = \frac{100!}{2^{100} \cdot (50!)^2} \approx \underline{0.0796} \end{aligned} \tag{2}$$

#5.17. Сперва посчитаем вероятность того, что B выиграет. Она равна $P(B) =$
$= P\left(\overline{A_1}\right) \cdot P(B|A_1)$, где $P\left(\overline{A_1}\right)$ – это вероятность того, что A не выигрывает в свой первый ход, а $P(B|A_1)$ – это вероятность того, что B выиграет в свой ход.

$$P\left(\overline{A_1}\right) = 1 - P(A_1) = 1 - 0.3 = 0.7 \Rightarrow P(B) = P\left(\overline{A_1}\right) \cdot P(B|A_1) = 0.7 \cdot 0.5 = \underline{0.35} \tag{3}$$

Теперь посчитаем вероятность победы A. Это возможно в двух случаях:

1. После первого хода. Вероятность победы равна $P(A_{\text{after 1}}) = 0.3$
2. После второго хода. Вероятность победы равна $P(A_{\text{after 2}}) = P\left(\overline{B}\right) \cdot P(A_2) = 0.35 \cdot 0.4 = 0.14$

Отсюда вероятность победы:

$$P(A) = P(A_{\text{after 1}}) + P(A_{\text{after 2}}) = 0.3 + 0.14 = \underline{0.44} \tag{4}$$

2. Problems to Model

#2.24. Выберем $n = 12$, $m = 7$, $k = 4$. Случайным образом распределим признаки для массива n_j и для массива m_j.

Вычисленное аналитическим способом результат равен $P_S = \frac{\prod_{j=1}^{4} C_{n_j}^{m_j}}{C_{12}^{7}} = \frac{27}{792} = 0.034(09)$.

Для моделирования использовался язык программирования Python. Исходный код программы представлен в листинге 1. Сперва создадим функцию, имитирующую выбор m изделий из массива n_j. И запустим 100000 экспериментов, из которых выделим благоприятные.

Листинг 1. Моделирование задачи 2.24

```python
from functools import reduce
from random import *
import math

def get_random_mj(_array_n, _m, _k):
    sublist = _array_n.copy()
    shuffle(sublist)
    sublist = sublist[:_m]
    return [sublist.count(i) for i in range(_k + 1)]

def equals(a, b):
    for index in range(1, len(a)):
        if a[index] != b[index]:
            return False
    return True

def product_of_list(list): return reduce(lambda x, y: x * y, list)

if __name__ == '__main__':
    n, m, k = 12, 7, 4

    array_n = [1, 4, 3, 2, 4, 3, 1, 2, 3, 2, 4, 1]
    nj = [array_n.count(i) for i in range(n + 1)]
    mj = [0, 2, 1, 3, 1]

    all_tries, good_tries = 1000000, 0
    for i in range(all_tries):
        random_mj = get_random_mj(array_n, m, k)
        if equals(random_mj, mj):
            good_tries += 1

    model = good_tries / all_tries
    print("Смоделированный результат: ", model)

    prod = product_of_list([math.comb(nj[i], mj[i]) for i in range(1, k + 1)])
    mCn = math.comb(n, m)
    solve = prod / mCn
    print("Вычисленный аналитическим способом результат: ", solve)

    print(f"Отношение результатов: {(model / solve):.2}")
```

Смоделированное решение равно $P_M = \frac{n_{\text{бл}}}{100000} = \underline{0.034173}$. С аналитическим решением совпадение в 2 значащие цифры.

#3.17. Выберем $T = 1000000$. Случайным образом зададим $t = 780014$ (можно и вручную задать, результат не изменится). Вычисленное аналитическим способом результат равен:

$$P_S = 1 - \left(1 - \frac{t}{T}\right)^2 = 1 - \left(1 - \frac{780014}{1000000}\right)^2 = 0.951606159804$$

Для моделирования использовался язык программирования Python. Исходный код программы представлен в листинге 2. Запустим 10000000 экспериментов со случайными точками (a, b). Из них выберем благоприятные исходы с помощью неравенства , полученного в аналитическом решении.

Листинг 2. Моделирование задачи 3.17

```python
from random import *

if __name__ == '__main__':
    T = 1000000
    t = randint(0, T)
    tries_number = 10000000
    all_tries = [(randint(0, T), randint(0, T)) for _ in range(tries_number)]
    good_tries = [(a, b) for (a, b) in all_tries if abs(a - b) <= t]

    model = len(good_tries) / len(all_tries)
    print("Смоделированный результат при t =", t, ":", model)

    solve = 1 - (1 - t / T) ** 2
    print("Вычисленный аналитическим способом результат: ", solve)

    print(f"Отношение результатов: {(model / solve):.2}")
```

Смоделированное решение равно $P_M = \frac{\text{good\_tries}}{10000000} = \frac{9515833}{10000000} = \underline{0.9515833}$. С аналитическим решением совпадение в 3 значащие цифры.

#4.32. Аналитическое решение представлено в выражении (2).
Для моделирования использовался язык программирования Python. Исходный код программы представлен в листинге 3. Создадим функцию, имитирующую вытаскивание шара из урны и возвращающую количество попыток для вытягивания черного шара. Если черный шар не извлекли, то возвращается -1. Запустим 1000000 экспериментов с помощью созданной функции *take_black_try()* и выберем благоприятные исходы.

Листинг 3. Моделирование задачи 4.32

```
from random import *

def take_black_try():
    _try = 1
    _bin = ['w', 'b']
    while True:
        if _try == 51:
            return -1
        shuffle(_bin)
        if _bin[0] == 'w':
            _bin.append('w')
            _bin.append('w')
            _try += 1
        else:
            return _try

if __name__ == '__main__':
    tries_number = 1000000
    good_tries = 0
    for i in range(tries_number):
        _try = take_black_try()
        if _try == -1:
            good_tries += 1

    model = good_tries / tries_number
    print("Смоделированный результат:", model)
```

Смоделированное решение равно $P_M = \frac{\text{good\_tries}}{1000000} = \frac{7943}{1000000} = \underline{0.07943}$. С аналитическим решением совпадение в 2 значащие цифры.

#5.17. Аналитическое решение представлено в выражении (3) и (4).
Для моделирования использовался язык программирования Python. Исходный код программы представлен в листинге 4. Создадим функцию, имитирующую игру из задания. Если побеждает A, то возвращается 'A', если побеждает B, то возвращается 'B', если никто не выигрывает, то возвращается 'N'. Запустим 10000000 экспериментов с помощью созданной функции *game()* и выберем благоприятные исходы для A и для B.

Листинг 4. Моделирование задачи 5.17

```
from random import *

def game():
    if randint(1, 10) <= 3:
        return 'A'
    if randint(1, 10) <= 5:
        return 'B'
    if randint(1, 10) <= 4:
        return 'A'
    return 'N'

if __name__ == '__main__':
    tries_number = 10000000
    a_wins, b_wins = 0, 0
    for i in range(tries_number):
        result = game()
        if result == 'A':
            a_wins += 1
        elif result == 'B':
            b_wins += 1
        else:
            continue

    model_a = a_wins / tries_number
    print("Смоделированный результат для A:", model_a)
    model_b = b_wins / tries_number
    print("Смоделированный результат для B:", model_b)
```

Смоделированное решение для A равно $P(A)_M = \frac{\text{a\_wins}}{10000000} = \underline{0.4400509}$, а для B равно $P(B)_M = \frac{\text{b\_wins}}{10000000} = \underline{0.3500742}$. С аналитическим решением совпадение в 2 значащие цифры.