



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«Крымский федеральный университет имени В.И. Вернадского»

Физико-технический институт

Кафедра компьютерной инженерии и моделирования

Лабораторная работа № 2

**«Количество информации при неполной достоверности
сообщений»**

по дисциплине

«Теория информации и кодирование»

Выполнил:
студент 3 курса
группа ИВТ-222
Гоголев В. Г.

Проверил:
Филиппов Д.М.
«___» _____ 20__ г.
Подпись: _____

Симферополь, 2024

Цель: рассчитать количество информации, приходящее к получателю, при наличии помех.

Техническое задание: на вход информационного устройства поступает совокупность дискретных сообщений $\{x_i\}$, где $i=1 \div N$. Вероятности появления дискретных сообщений на входе задаются в виде счетчика случайных чисел. Вероятности безошибочной передачи сообщения составляют не менее 70%. Вероятности безошибочной передачи задаются случайным образом. Вероятности ошибок генерируются счетчиком случайных чисел. Необходимо разработать программное обеспечение и провести комплекс численных экспериментов по расчету количества информации, получаемое при неполной достоверности сообщений.

Ход работы:

Вариант № 4

С использованием разработанного программного обеспечения необходимо провести комплекс численных экспериментов (не менее 6), в ходе которого необходимо:

- а) сгенерировать массив вероятностей появления совокупности дискретных сообщений на входе информационного устройства ($P(X)$);
- б) сгенерировать матрицу вероятностей перехода со входа на выход ($P(X/Y)$);
- в) рассчитать вероятности появления совокупности дискретных сообщений на выходе информационного устройства ($P(Y)$);
- г) рассчитать матрицу вероятностей совместных событий ($P(X,Y)$);
- д) определить энтропию на входе информационного устройства ($H(X)$);
- е) определить остаточную или условную энтропию выходного сообщения относительно входного ($H(X/Y)$);

ж) определить количество информации при неполной достоверности сообщений ($I(X, Y)$).

а) сгенерировать массив вероятностей появления совокупности дискретных сообщений на входе информационного устройства ($P(X)$);

```
def generate_P_X(N):  
    P_X = [random.random() for _ in range(N)]  
    P_X = [p / sum(P_X) for p in P_X]  
    return P_X
```

Рисунок 1 – программная реализация функции для генерации массива входных сообщений

По факту используется функция из первой лабораторной работы, которая принимает на вход (N) число сообщений и возвращает одномерный массив со случайными вероятностями сообщений.

б) сгенерировать матрицу вероятностей перехода со входа на выход ($P(X/Y)$);

```
def generate_matrix_P_X_Y(N):  
    P_X_Y = []  
    for i in range(N):  
        row = [(random.random()) for _ in range(N)]  
        row[i] = (random.uniform(0.7, 1))  
  
        total = sum(row) - row[i]  
        if total > 0:  
            for j in range(N):  
                if j != i:  
                    row[j] = row[j] / total * (1 - row[i])  
  
        P_X_Y.append(row)  
    return P_X_Y
```

Рисунок 2 – функция, реализующая матрицу перехода со входа на выход устройства

Реализована согласно технического задания, числа задаются случайно, но вероятность безошибочной передачи составляет $> 70\%$, значит, что диагональные элементы матрицы: $[1,1], [2,2] \dots [k,k]$ должны иметь вероятность больше чем 0.7.

Далее в функции происходит нормализация элементов матрицы для того, чтобы сумма элементов в строке была равна единице.

Функция принимает N (кол-во сообщений) и возвращает матрицу $P(x/y)$

в) рассчитать вероятности появления совокупности дискретных сообщений на выходе информационного устройства ($P(Y)$).

На выходе информационного устройства мы так же получаем массив вероятностей $P(y)$, элементы которого рассчитываются по данной формуле:

$$p(y_j) = \sum_{i=1}^N p(x_i) \cdot p(x_i / y_j)$$

```
def calculate_P_Y(P_X_Y, P_X):
    P_Y = [0] * len(P_X_Y)
    for j in range(len(P_X_Y)):
        for i in range(len(P_X_Y)):
            P_Y[j] += P_X[i] * P_X_Y[i][j]
    return P_Y
```

Рисунок 3 – функция, реализующая массив вероятностей выходных сообщений

г) рассчитать матрицу вероятностей совместных событий ($P(X,Y)$).

Каждый элемент матрицы вероятности совместных событий, рассчитывается согласно формуле из методических указаний:

$$p(x_i, y_j) = p(y_j) \cdot p(x_i / y_j)$$

```
def calculate_matrix_P_XY(P_X, P_X_Y):  
    P_XY = [[0] * len(P_X_Y) for _ in range(len(P_X))]  
    for i in range(len(P_X)):  
        for j in range(len(P_X_Y)):  
            P_XY[i][j] = P_X[i] * P_X_Y[i][j]  
    return P_XY
```

Рисунок 4 – функция, рассчитывающая матрицу вероятностей совместных событий

д) определить энтропию на входе информационного устройства ($H(X)$).

Формула для вычисления энтропии на входе:

$$H(X) = - \sum_{i=1}^N p(x_i) \log_2 p(x_i)$$

```
def calculate_entropy(P_X):  
    entropy = 0  
    for p in P_X:  
        entropy -= p * math.log2(p)  
    return entropy
```

Рисунок 5 – функция для вычисления энтропии на входе устройства

е) определить остаточную или условную энтропию выходного сообщения относительно входного ($H(X/Y)$).

Формула для вычисления условной энтропии использована из методических указаний:

$$H(X/Y) = - \sum_{i=1}^N \sum_{j=1}^N p(x_i, y_j) \log_2 p(x_i / y_j)$$

```
def calculate_conditional_entropy(P_XY, P_X_Y):  
    H_X_Y = 0  
    for i in range(len(P_XY)):  
        for j in range(len(P_XY[i])):  
            H_X_Y -= P_XY[i][j] * math.log2(P_X_Y[i][j])  
    return H_X_Y
```

Рисунок 6 – функция, вычисляющая условную энтропию

ж) определить количество информации при неполной достоверности сообщений ($I(X,Y)$).

Формула для вычисления кол-ва информации при неполной достоверности сообщений:

$$I(X,Y) = H(X) - H(X/Y)$$

```

вероятности входных сообщений
0.261304
0.157066
0.342846
0.020729
0.218054
матрица вероятностей перехода со входа на выход (P(X/Y))
0.909089 0.038254 0.025978 0.012075 0.014604
0.081894 0.738865 0.081723 0.063574 0.033945
0.053511 0.014664 0.848052 0.060210 0.023562
0.014324 0.004000 0.020523 0.940216 0.020938
0.022920 0.016131 0.012682 0.006727 0.941540
вероятности появления совокупности дискретных сообщений на выходе информационного устройства P(Y)
0.274053
0.134674
0.313566
0.054740
0.222966
матрица вероятностей совместных событий (P(X,Y))
0.237549 0.009996 0.006788 0.003155 0.003816
0.012863 0.116050 0.012836 0.009985 0.005332
0.018346 0.005028 0.290752 0.020643 0.008078
0.000297 0.000083 0.000425 0.019490 0.000434
0.004998 0.003518 0.002765 0.001467 0.205307
Энтропия на входе информационного устройства(H(X)): 2.05 бит
Остаточная энтропия выходного сообщения относительно входного(H(X/Y)): 0.78 бит
Количество информации при неполной достоверности сообщения: 1.27 бит

```

Рисунок 7 – пример вывода одного из экспериментов (для N= 5)

```

Энтропия на входе информационного устройства(H(X)): 5.50 бит
Остаточная энтропия выходного сообщения относительно входного(H(X/Y)): 1.48 бит
Количество информации при неполной достоверности сообщения: 4.02 бит
-----
Среднее количество информации, получаемое при неполной достоверности сообщений в ходе 6 экспериментов: 0.69 бит

```

Рисунок 8 – вывод основных параметров одного из экспериментов (N = 58)

Задание II. Рассчитать среднее количество информации, получаемое при неполной достоверности сообщений в ходе проведенных экспериментов.

Среднее количество информации, получаемое при неполной достоверности сообщений в ходе проведенных экспериментов считается как сумма кол-ва информации за 6 экспериментов/кол-во экспериментов.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы были получены навыки рассчитывать количество информации, приходящее к получателю, при наличии помех.

Изучены понятия совместной вероятности, условной вероятности, построены матрицы условных вероятностей, совместных вероятностей. Введен термин условной энтропии, количества информации при неполной достоверности сообщения.

ПРИЛОЖЕНИЕ

```
import math
import random
import pandas as pd

N = 5

#а) сгенерировать массив вероятностей появления совокупности дискретных
сообщений на входе информационного устройства ( $P(X)$ )
def generate_P_X(N):
    P_X = [random.random() for _ in range(N)]
    P_X = [p / sum(P_X) for p in P_X]
    return P_X

#б) сгенерировать матрицу вероятностей перехода со входа на выход ( $P(X/Y)$ )
(матрица условных вероятностей)
def generate_matrix_P_X_Y(N):
    P_X_Y = []
    for i in range(N):
        row = [(random.random()) for _ in range(N)]
        row[i] = (random.uniform(0.7, 1))

        total = sum(row) - row[i]
        if total > 0:
            for j in range(N):
                if j != i:
                    row[j] = row[j] / total * (1 - row[i])

        P_X_Y.append(row)
    return P_X_Y

#в) рассчитать вероятности появления совокупности дискретных сообщений на выходе
информационного устройства  $P(Y)$ 
def calculate_P_Y(P_X_Y, P_X):
    P_Y = [0] * len(P_X_Y)
    for j in range(len(P_X_Y)):
        for i in range(len(P_X_Y)):
            P_Y[j] += P_X[i] * P_X_Y[i][j]
    return P_Y

#г) рассчитать матрицу вероятностей совместных событий ( $P(X,Y)$ )
def calculate_matrix_P_XY(P_X, P_X_Y):
    P_XY = [[0] * len(P_X_Y) for _ in range(len(P_X))]
    for i in range(len(P_X)):
        for j in range(len(P_X_Y)):
            P_XY[i][j] = P_X[i] * P_X_Y[i][j]
    return P_XY

#д) определить энтропию на входе информационного устройства ( $H(X)$ )
```

```

def calculate_entropy(P_X):
    entropy = 0
    for p in P_X:
        entropy -= p * math.log2(p)
    return entropy

#е)определить условную энтропию выходного сообщения относительно входного
(H(X/Y));
def calculate_conditional_entropy(P_XY,P_X_Y):
    H_X_Y = 0
    for i in range(len(P_XY)):
        for j in range(len(P_XY[i])):
            H_X_Y -= P_XY[i][j] * math.log2(P_X_Y[i][j])
    return H_X_Y

#ж)определить количество информации при неполной достоверности сообщений (I(X,Y))
def calculate_count_inf(H_X,H_X_Y):
    return H_X-H_X_Y

I_avg = 0
experement_count = 6

for i in range(experement_count):

    print(f"-----ТЕСТ-№{i+1}-----")
    P_X = generate_P_X(N)
    P_X_Y = generate_matrix_P_X_Y(N)
    P_Y = calculate_P_Y(P_X_Y, P_X)
    P_XY = calculate_matrix_P_XY(P_X,P_X_Y)
    H_X = calculate_entropy(P_X)
    H_X_Y = calculate_conditional_entropy(P_XY,P_X_Y)
    I_X_Y = calculate_count_inf(H_X,H_X_Y)

    print("вероятности входных
сообщений\n",pd.DataFrame(P_X).to_string(index=False, header=False))
    print("матрица вероятностей перехода со входа на выход
(P(X/Y))\n",pd.DataFrame(P_X_Y).to_string(index=False, header=False))
    print("вероятности появления совокупности дискретных сообщений на выходе
информационного устройства P(Y)\n",pd.DataFrame(P_Y).to_string(index=False,
header=False))
    print("матрица вероятностей совместных событий
(P(X,Y))\n",pd.DataFrame(P_XY).to_string(index=False, header=False))
    print(f"Энтропия на входе информационного устройства(H(X)): {H_X:.2f} бит")
    print(f"Остаточная энтропия выходного сообщения относительно входного(H(X/Y)):
{H_X_Y:.2f} бит")
    print(f"Количество информации при неполной достоверности сообщения:
{I_X_Y:.2f} бит")

    I_avg += I_X_Y/experement_count

```

```
    print(f"Среднее количество информации, получаемое при неполной  
достоверности сообщений в ходе {experiment_count} экспериментов:  
{I_avg/experiment_count:.2f} бит")  
  
    I_avg += I_X_Y/experiment_count  
  
    print(f"Среднее количество информации, получаемое при неполной  
достоверности сообщений в ходе {experiment_count} экспериментов:  
{I_avg/experiment_count:.2f} бит")
```

Приложение 1 – листинг программного кода