**Национальный исследовательский технологический университет «МИСИС»**

**Институт Компьютерных Наук (ИКН)**

**Курс «Основы теории информации»**

Лабораторная работа № 1

**«Решение задачи 1.3»**

Выполнил: студент группы БИВТ-22-СП-5

Изображение выглядит как зарисовка

Автоматически созданное описаниеШманёв Е. А.

Проверил: Куприянов В.В.

Москва, 2024

**Условие задачи:**

Пусть опыт Y состоит в извлечении одного шара из урны, содержащей 5 чёрных и 10 белых шаров, опыт Xk - в предварительном извлечении из той же урны (без возвращения обратно) K шаров. Чему равно энтропия опыта Y и количество информации об этом опыте, содержащейся в опытах X1, X2, X13, X14?

**Решение:**

Эта задача требует расчета энтропии опыта Y*Y* и информации *I*(*Y*;*Xk*​), где *Xk*​ - это опыт, состоящий в извлечении *k* шаров до извлечения *Y*. Основные шаги решения:

1. Вычислить вероятность *P*(*Y*) (извлечения чёрного или белого шара) без предварительных опытов.
2. Рассчитать энтропию опыта *Y* по формуле: *H*(*Y*)=−*i*∑​*P*(*Yi*​)log2​(*P*(*Yi*​)).
3. Для каждого *k* (1, 2, 13, 14):
   * Вычислить условные вероятности *P*(*Y*;*Xk*​) для чёрного и белого шара.
   * Вычислить энтропию *H*(*Y*;*Xk*​).
   * Найти количество информации *I*(*Y*;*Xk*​) как разность:

*I*(*Y*;*Xk*​)=*H*(*Y*)−*H*(*Y*;*Xk*​).

**Код:**

import math

from itertools import combinations

# Функция для вычисления энтропии

def entropy(probabilities):

    return -sum(p \* math.log2(p) for p in probabilities if p > 0)

# Общие данные

total\_balls = 15

black\_balls = 5

white\_balls = 10

# Вероятности для Y (без предварительных извлечений)

P\_black = black\_balls / total\_balls

P\_white = white\_balls / total\_balls

# Энтропия опыта Y

H\_Y = entropy([P\_black, P\_white])

# Функция для вычисления условных вероятностей и условной энтропии

def conditional\_entropy(k):

    # Суммирование по всем возможным комбинациям извлечений

    H\_Y\_given\_Xk = 0

    for blacks\_drawn in range(max(0, k - white\_balls), min(k, black\_balls) + 1):

        whites\_drawn = k - blacks\_drawn

        # Вероятность этой комбинации извлечений

        P\_Xk = (math.comb(black\_balls, blacks\_drawn) \*

                math.comb(white\_balls, whites\_drawn) /

                math.comb(total\_balls, k))

        # Обновленные вероятности для Y после Xk

        remaining\_black = black\_balls - blacks\_drawn

        remaining\_white = white\_balls - whites\_drawn

        remaining\_total = total\_balls - k

        if remaining\_total > 0:

            P\_Y\_black\_given\_Xk = remaining\_black / remaining\_total

            P\_Y\_white\_given\_Xk = remaining\_white / remaining\_total

            # Энтропия для данной комбинации

            H\_Y\_given\_Xk += P\_Xk \* entropy([P\_Y\_black\_given\_Xk, P\_Y\_white\_given\_Xk])

    return H\_Y\_given\_Xk

# Считаем информацию для X1, X2, X13, X14

results = {}

for k in [1, 2, 13, 14]:

    H\_Y\_given\_Xk = conditional\_entropy(k)

    I\_Y\_Xk = H\_Y - H\_Y\_given\_Xk

    results[k] = (H\_Y\_given\_Xk, I\_Y\_Xk)

H\_Y, results

**Результат:**

* Чем больше извлеченных шаров (*k*), тем больше информации об опыте *Y*.
* При *k*=14, опыт *X*14​ полностью определяет результат Y*Y*, давая максимальную информацию 0.918бит, и условная энтропия становится равной 0.

**{1: (0.9145641619692977, 0.0037316720851918728),**

**2: (0.9102259771088079, 0.00806985694568163),**

**13: (0.47619047619047616, 0.4421053578640134),**

**14: (0.0, 0.9182958340544896)}**