

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Дополнительное задание к лабораторной работе №1
по дисциплине «Статистическое моделирование»

Выполнил студент
гр. 33534/5

Стойкоски Н.С.

Руководитель

Чуркин В.В.

Содержание

Цель работы	3
Проведение работы	3
Результаты	4
Вывод.....	4
Текст программы.....	5

Цель работы

Дополнительное задание к лабораторной работе №1 состоит в выполнении программно тест оценки статистических свойств применяемого генератора случайных чисел.

Вариант 10. Проверка перестановок:

Данный тест проверяет равномерность распределения символов в исследуемой последовательности, анализируя взаимное расположение чисел в подпоследовательностях. Полученный результат анализируется при помощи критерия хи-квадрат с числом степеней свободы, равным $t! - 1$

Проведение работы

Пусть $\varepsilon = \varepsilon_1\varepsilon_2...\varepsilon_n$ – последовательность m -разрядных чисел длины n . Разобьем ее на подпоследовательности длиной t чисел каждая (лишние биты отбрасываются). В каждой такой подпоследовательности возможно $t!$ вариантов относительного расположения чисел. Подсчитывается, сколько раз встречается каждое такое расположение V_i , $i = 1..t!$ и вычисляется статистика:

$$\chi^2(obs) = \sum_{i=1}^{t!} \frac{\left(v_i - \left[\frac{n}{t} \right] \cdot \frac{1}{t!} \right)^2}{\left[\frac{n}{t} \right] \cdot \frac{1}{t!}} = \frac{1}{\left[\frac{n}{t} \right] \cdot \frac{1}{t!}} \sum_{i=1}^{t!} \left(v_i - \left[\frac{n}{t} \right] \cdot \frac{1}{t!} \right)^2 .$$

Полученный результат анализируется при помощи критерия хи-квадрат с числом степеней свободы, равным $t! - 1$.

Написана функция на языке python которая генерирует последовательность $n=10000$ равномерно распределенных чисел на интервале $[0, 1]$. Полученная последовательность разделяется на n/t подпоследовательностей, каждой из которых отдельно рассматривается – каждому элементу сопоставляется индекс $0..t-1$, а потом подпоследовательность сортируется по возрастанию сгенерированных чисел. В результате имеем относительное расположение представленное индексами после сортировки. Для каждое найденное расположение, в словарь храним число – сколько раз найдено такое расположение. После обработки всей последовательности, вычисляем статистику.

Результаты

Уровень значимости $\alpha = 0.05$

t	N	χ^2	Критическое значение
2	1	0.051	3,841
3	5	2.155	11,07
4	23	23.36	35,17
5	119	120.04	145,46
6	719	667.46	782,49
7	5039	5148.76	18014

При уровень значимости $\alpha = 0.05$ гипотеза H_0 принимается.

Вывод

В ходе лабораторной работы был выполнен программный тест оценки статистических свойств применяемого генератора случайных чисел. Была выполнена проверка перестановок - проверка равномерность распределения в исследуемой последовательности, анализируя взаимное расположение чисел. Полученный результат был анализирован при помощи критерия хи-квадрат. В результате нулевая гипотеза принимается с уровнем значимости 0.05.

Текст программы

```
import numpy as np
import random
import math
from collections import Counter

def permutation_test(t, n):
    u = [random.random() for _ in range(n)]
    numIntervals = int(n / t)
    combCount = Counter()
    for i in range(numIntervals):
        sublist = [(u[i*t + j], j) for j in range(t)]
        sublist.sort(key=lambda v: v[0])
        combinationId = 0
        for j in range(t):
            combinationId += sublist[j][1]*(10**j)
        combCount[combinationId] += 1

    v = [x[1] for x in combCount.items()]
    t_fact = math.factorial(t)
    num_zero_combs = t_fact - len(v)
    term = (n/t)*(1/t_fact)
    sum = np.sum([(vi - term)**2 for vi in v])
    sum += num_zero_combs * (-(n/t)*(1/t_fact))**2
    chi2 = (1/((n/t)*(1/t_fact)))*sum
    return chi2

print('t, N, chi2')
for t in range(2, 20, 1):
    t_fact = math.factorial(t)
    N = t_fact - 1
    chi2 = permutation_test(t, 10**4)
    print(t, ' ', N, ' ', chi2)
```