МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа № 2**

Дисциплина: Теория надежности

Выполнил: студент группы ВТ-31

Ковалёв И. Д.

Проверил:

Кабалянц П. С

Белгород 2020

Пьяница стоит между двумя пропастями, с одной стороны река, с другой копья. В начальный момент времени пьяница стоит на обеих ногах, из этого положения он может встать на левую ногу, на правую или остаться на месте. С левой ноги он может упасть в реку или встать на обе ноги, С правой ноги может упасть на копья или встать на обе ноги.

Составим матрицу переходных вероятностей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Р | К | Л | О | П |
| Р | 1 | 0 | 0 | 0 | 0 |
| К | 0 | 1 | 0 | 0 | 0 |
| Л | 0.23 | 0 | 0 | 0.77 | 0 |
| О | 0 | 0 | 0.14 | 0.45 | 0.41 |
| П | 0 | 0 | 0.47 | 0.53 | 0 |

Проведем эксперимент 1000 раз:

Среднее количество шагов за каждую реинкарнацию любителя спиртного: 6.96

При этом упал в реку 144 раз, а на копья 856 раз.

Сравним с теоретическими значениями.

– среднее время пребывания в , если начали с

|  |  |  |
| --- | --- | --- |
| 1.47 | 3.42 | 1.40 |
| 0.60 | 4.44 | 1.82 |
| 0.32 | 2.34 | 1.96 |

N =

– вероятность поглотиться в , если начали в

|  |  |
| --- | --- |
| 0.34 | 0.66 |
| 0.14 | 0.86 |
| 0.07 | 0.93 |

B =

Должен был упасть в реку 139 раз, а на копья 860 раз.

Среднее количество шагов: 6.86.

Проверим гипотезу о том, что экспериментальная вероятность поглотиться в (река) соответствует теоретической вероятности.

# This Python file uses the following encoding: utf-8

import random

import math

import numpy

from prettytable import PrettyTable

import matplotlib.pyplot as pyplot

i = 3

j = 10

k = 9

lr = i / (i + j)

lo = j / (i + j)

ol = i / (i + j + k)

oo = j / (i + j + k)

op = k / (i + j + k)

po = j  / (j + k)

pk = k / (j + k)

p0 = [0, 0, 0, 1, 0]

# Матрица переходных вероятностей

#     Р   К   Л   О   П

P = [[1,  0,  0,  0,  0 ], # Р

     [0,  1,  0,  0,  0 ], # К

     [lr, 0,  0,  lo, 0 ], # Л

     [0,  0,  ol, oo, op], # О

     [0,  pk, 0,  po, 0 ]] # П

Q = [[0,  lo, 0 ],

     [ol, oo, op],

     [0,  po, 0 ]]

R = [[lr, 0],

     [0, 0],

     [0, pk]]

# Вероятности перехода из одного положения в другое: из текущего / в текущее

stepProbs = [[0,  0,  lr],

             [lr, 0,  lo],

             [ol, oo, op],

             [po, 0,  pk],

             [0,  0,  pk]]

def oneDrunkStep(currentState):

     # Возвращает новое состояние любителя спиртного

     if currentState == 0 or currentState == 4:

          return currentState # Из реки и копий уже не выберешься

     r = random.random()

     if r < stepProbs[currentState][0]:

          return currentState - 1

     if r >  stepProbs[currentState][0] and r < stepProbs[currentState][1] + stepProbs[currentState][0]:

          return currentState

     else:

          return currentState + 1

if \_\_name\_\_ == '\_\_main\_\_':

     print("Матрица переходных вероятностей:")

     print(P)

     N = (numpy.eye(3) - Q)

     N = numpy.linalg.inv(N)

     B = N @ R

     r = 0               # Сколько раз упал в реку

     s = 0               # Сколько раз упал на копья

     k = 0               # Количество шагов за одну жизнь

     averageSteps = 0    # Среднее количество шагов

     state = 2           # Текущее положение тунеядца, алкоголика и дебошира

     sz = 1000        # Сколько раз неприкаянная душа любителя качественно выпить и грамотно закусить будет возвращатся в его бренное тело, стоящее на левой ноге

     for i in range (0, sz):

          while state != 0 and state != 4:

               state = oneDrunkStep(state)

               k += 1

          if state == 4:

               s += 1

          else:

               r += 1

          averageSteps += k

          k = 0

          state = 2

     averageSteps /= sz

     print("Проведем эксперимент {} раз".format(sz))

     print("Среднее количество шагов за каждую реинкарнацию любителя спиртного: {}".format(round(averageSteps, 2)))

     print("При этом упал в реку {} раз, а на копья {} раз".format(r, s))

     print("Сравним с теоретическими значениями.")

     print("Должен был упасть в реку {} раз, а на копья {} раз.".format(int(B[1][0]\*sz), int(B[1][1]\*sz)))

     print("Cреднее количество шагов: {}.".format(round(numpy.sum([N[1][i] for i in range(0, 3)]), 2)))

     K\_t = 1.64     # 6 степеней свободы

     print("Проверка гипотезы о том, что экспериментальная вероятность поглотиться в S\_0 (река), начиная с S\_3 (обе ноги) соотвествует теоретическому значению:")

     p0 = math.sqrt(B[2][0])

     K\_v = (r / sz - p0) / (math.sqrt(p0 \* (1 - p0) / 6))

     print("К\_в = {}, К\_т: {}\n{}".format(round(K\_v, 2), K\_t, "Гипотеза верна!" if abs(K\_v) < K\_t else "Гипотезу следует отбросить!"))