МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 1

по курсу «**Случайные процессы**»

Тема: **Однородная цепь Маркова с тремя состояниями**

Выполнил:
Студент 4-го курса
Гогинян Б.А.
Группа: КМБО-03-16

МОСКВА 2019

Лабораторная работа по случайным процессам № 1

«Однородная цепь Маркова с 3-мя состояниями»

Дана матрица переходных вероятностей однородной цепи Маркова

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix}$$

Построить граф состояний цепи Маркова.

Найти:

1. Матрицы переходных вероятностей за n шагов P^n $(n = 2,..., n_{\min})$ и величины отклонений $\delta_n = \max(|p_{ij}(n) - p_{ij}(n-1)|; i, j = 1,2,3)$ (для $n = 2,..., n_{\min}$), где $n_{\min} = \min(n | \delta_n < 0{,}00001)$. Результаты представить в табличной форме:

n	P^n	δ_n
1	$\begin{pmatrix} p_{11}(1) & p_{12}(1) & p_{13}(1) \\ p_{21}(1) & p_{22}(1) & p_{23}(1) \\ p_{31}(1) & p_{32}(1) & p_{33}(1) \end{pmatrix}$	–
2	$\begin{pmatrix} p_{11}(2) & p_{12}(2) & p_{13}(2) \\ p_{21}(2) & p_{22}(2) & p_{23}(2) \\ p_{31}(2) & p_{32}(2) & p_{33}(2) \end{pmatrix}$	δ_2
3	$\begin{pmatrix} p_{11}(3) & p_{12}(3) & p_{13}(3) \\ p_{21}(3) & p_{22}(3) & p_{23}(3) \\ p_{31}(3) & p_{32}(3) & p_{33}(3) \end{pmatrix}$	δ_3
…	…	…
k	$\begin{pmatrix} p_{11}(k) & p_{12}(k) & p_{13}(k) \\ p_{21}(k) & p_{22}(k) & p_{23}(k) \\ p_{31}(k) & p_{32}(k) & p_{33}(k) \end{pmatrix}$	δ_k

где $k = n_{\min}$.

2. Стационарное распределение вероятностей состояний (r_1, r_2, r_3). Провести проверку стационарности найденного распределения.

3. Распределения вероятностей состояний через n шагов $(p_1(n), p_2(n), p_3(n))$ (для $n=1,..., m_{\min}$) и величины отклонений $\delta_n = \max(|p_i(n) - r_i|; i=1,2,3)$ (для $n=1,..., m_{\min}$), где $m_{\min} = \min(n | \delta_n < 0{,}00001)$, для следующих начальных распределений: $(1,0,0)$, $(0,1,0)$ и $(0,0,1)$. Результаты представить в табличной форме:

n	$(p_1(n), p_2(n), p_3(n))$	δ_n
0	$(1,0,0)$	$\delta_0 = \max((1-r_1), r_2, r_3)$
1		δ_1
2		δ_2
…		…
k		δ_k

где $k = m_{\min}$. Аналогично для $(0,1,0)$ и $(0,0,1)$.

4. Для каждого состояния $i=1,\ 2,\ 3$, взятого в качестве начального $i=i_0$ провести в соответствии с матрицей переходных вероятностей генерацию последовательности номеров состояний $i_1,...,i_n$ через n шагов, определяя для каждого n значения $R(i,n) = |\{i_k = i;\ k=1,...,n\}|$ (число возвратов в состояние i) и $\nu(i,n) = \dfrac{R(i,n)}{n}$ (число возвратов в состояние i). Генерацию проводить до шага $N_{\min}(i) = \min(n | \Delta_n(i) < 0{,}001)$, где $\Delta_n(i) = |\nu(i,n) - r_i|$.

В отчете привести значения $N_{\min}(i)$, $i=1,\ 2,\ 3$.

5. По результатам пункта 4 для каждого начального состояния $i=1,\ 2,\ 3$ построить таблицы вида

n	$R(i,n)$	$\nu(i,n)$	$\Delta_n(i)$
1			
2			
…			
10			
$k-5$			
…			
k			

где $k = \max(16, N_{\min}(i))$.

Вычисления и вывод результатов проводить с точностью до 0,00001 .

Краткие теоретические сведения

<u>Опр.</u> Последовательность случайных величин $\{X_n\}_{n=0}^{\infty}$ называется **цепью Маркова**, если для произвольного набора $i_1 < i_2 < ... < i_k$ $(k = 3,\ 4, ...)$ и любых $E_{j_1}, ..., E_{j_k}$ справедливо
$P(X_{i_k} = E_{j_k} | X_{i_1} = E_{j_1}, ..., X_{i_{k-1}} = E_{j_{k-1}}) = P(X_{i_k} = E_{j_k} | X_{i_{k-1}} = E_{j_{k-1}})$.
<u>Опр.</u> Цепь Маркова $\{X_n\}_{n=0}^{\infty}$ называется **однородной**, если для всех i и j вероятности $p_{ij} = P(X_{n+1} = E_j | X_n = E_i)$ не зависят от n.

<u>Опр.</u> Если существует $lim\,\overline{p}(n) = \overline{p}(\infty)$ при $n \to \infty$ и $\sum_j q_j = 1$.

<u>Опр.</u> Распределение $\overline{p^*}$ цепи Маркова называется стационарным, если оно остается неизменным на каждом шаге. Стационарное распределение в силу теоремы 1.2 ($\overline{p}(k + \overline{n}) = \overline{p}(k) \cdot P^n$) удовлетворяет соотношению $\overline{p^*} = \overline{p^*} \cdot P$

В однородной цепи Маркова вероятности p_{ij} называются переходными, а матрица $P = \|p_{ij}\|$ - матрицей переходных вероятностей цепи Маркова.
Матрица P обладает следующими свойствами:

1) $p_{ij} \geq 0$
2) $\sum_{j=1}^{N} p_{ij} = 1$ для всех $i = 1, 2, ..., N$.

Матрица, удовлетворяющая этим свойствам называется стохастической.

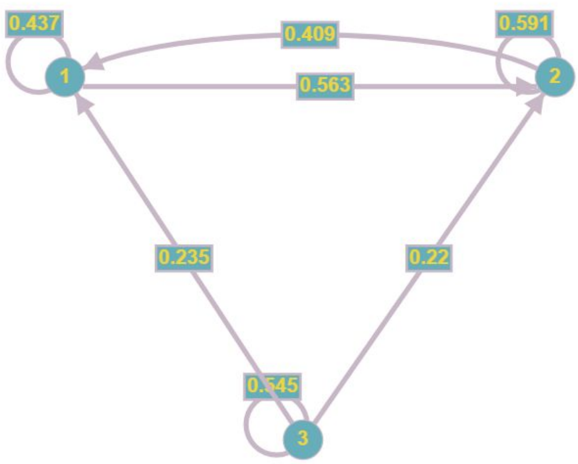
Средства языка программирования Python, которые использованы в программе расчета:

np.dot(A, B) - умножение матриц A и B
np.max(A) - находит максимальный элемент в матрице
np.abs(A) - модуль
np.random.random_sample() - возвращает случайное число из полуинтервала [0,1)

Результаты расчетов

Исходные данные:
Матрица переходных вероятностей однородной цепи Маркова
[0.437, 0.563, 0],
[0.409, 0.591, 0],
[0.235, 0.22, 0.545]

Граф состояний цепи Маркова:



1. Матрицы переходных вероятностей за n шагов P^n и величины отклонений δ_n :

n	P^n	δ_n
1	[0.437 0.563 0.] [0.409 0.591 0.] [0.235 0.22 0.545]	---
2	[0.42124 0.57876 0.] [0.42045 0.57955 0.] [0.32075 0.38222 0.29703]	0.13515
3	[0.42079 0.57921 0.] [0.42077 0.57923 0.] [0.3663 0.47182 0.16188]	0.073654774375
4	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.39109 0.52069 0.08822]	0.040141852
5	[0.42078 0.57922 0.] [0.42078 0.57922 0.]	0.0218773

	[0.4046 0.54732 0.04808]	
6	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.41196 0.56183 0.0262]	0.0119231336
7	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.41598 0.56974 0.01428]	0.0064981
8	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.41816 0.57405 0.00778]	0.0035414688
9	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.41935 0.5764 0.00424]	0.0019301
10	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42 0.57768 0.00231]	0.0010519
11	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42036 0.57838 0.00126]	0.000573288
12	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42055 0.57876 0.00069]	0.000312442
13	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42066 0.57897 0.00037424]	0.00017028
14	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42071 0.57908 0.00020396]	0.000092803
15	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42074 0.57914 0.00011116]	0.000050577683
16	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.42076 0.57918 0.000060582]	0.000027564837
17	[0.42078 0.57922 0.] [0.42078 0.57922 0.] [0.420771 0.57920 0.0000330172]	0.000015022836
18	[0.42078 0.57922 0] [0.42078 0.57922 0] [0.42078 0.57921 0.00002]	0.00000818744579

2. Найти стационарное распределение вероятностей состояний (r_1, r_2, r_3). Провести проверку стационарности найденного распределения.

(r_1, r_2, r_3) = (409/972, 563/972, 0) = (0.42078189, 0.57921811, 0)

Проверка:
$\overline{r} \cdot P = (0.42078189,\ 0.57921811,\ 0) \cdot P = (0.42078189,\ 0.57921811,\ 0) = \overline{r}$

3. Распределения вероятностей состояний через n шагов $(p_1(n), p_2(n), p_3(n))$ и величины отклонений δ_n, для следующих начальных распределений: (1,0,0), (0,1,0) и (0,0,1).

n	$(p_1(n), p_2(n), p_3(n))$	δ_n
0	(1,0,0)	0.5792181
1	(0.437, 0.563, 0)	0.0162181
2	(0.421236, 0.578764, 0)	0.0004541
3	(0.42079461, 0.57920539, 0)	0.0000127
4	(0.42078225, 0.57921775, 0)	0.00000035601

n	$(p_1(n), p_2(n), p_3(n))$	δ_n
0	(0,1,0)	0.420781893
1	(0.409, 0.591, 0)	0.011781893
2	(0.42045, 0.57955, 0)	0.000329893
3	(0.42077266, 0.57922734, 0)	0.000009237

n	$(p_1(n), p_2(n), p_3(n))$	δ_n
0	(0, 0, 1)	1.0
1	(0.235, 0.22, 0.545)	0.545

2	(0.32075, 0.382225, 0.297025)	0.29703
3	(0.36629865, 0.47182272, 0.16187863)	0.16189
4	(0.39108948, 0.52068667, 0.08822385)	0.08822
5	(0.40459956, 0.54731845, 0.048082)	0.04808
6	(0.41196252, 0.56183279, 0.02620469)	0.0262
7	(0.41597533, 0.56974311, 0.01428156)	0.01428
8	(0.41816232, 0.57405423, 0.00778345)	0.00778
9	(0.41935423, 0.5764038, 0.00424198)	0.00424
10	(0.42000381, 0.57768431, 0.00231188)	0.00231
11	(0.42035784, 0.57838219, 0.00125997)	0.00126
12	(0.42055078, 0.57876253, 0.00068669)	0.000687
13	(0.420656, 0.578969818, 0.000374244)	0.000374
14	(4.20713248e-01, 5.79082789e-01, 2.03962831e-04)	0.000204
15	(4.20744481e-01, 5.79144359e-01, 1.11159743e-04)	0.000112
16	(4.20761504e-01, 5.79177914e-01, 6.05820600e-05)	0.00006
17	(4.20770781e-01, 5.79196202e-01, 3.30172227e-05)	0.000033
18	(4.20775837e-01, 5.79206169e-01, 1.29943864e-05)	0.000017994
19	(0.420778592, 0.579211601, 0.00000980694056)	0.00000980694

4. $N_{min}(i)$ для каждого начального состояния i = 1, 2, 3:

$N_{min}(1) = 18$
$N_{min}(2) = 67$
$N_{min}(3) = 1$

Анализ результатов и выводы

(r_1, r_2, r_3)	P^k	$(p_1(n), p_2(n), p_3(n))$
(0.42078, 0.57922, 0)	[0.42078, 0.57921, 0] [0.42078, 0.57921, 0] [0.42077, 0.57919, 0.00001]	(0.42079, 0.57921, 0) (0.42045, 0.57955, 0) (0.42078, 0.57921, 0.00001)

Таблица №1 для нач. случая i=1

n	$R(i,\ n)$	$\nu(i,\ n)$	$\Delta_n(i)$
1	0	0	0.420078
2	1	0.5	0.07922
3	2	0.66667	0.24588
4	3	0.75	0.32922
5	3	0.6	0.17922
6	3	0.5	0.07922
7	3	0.42857	0.00779
8	3	0.375	0.04578
9	3	0.33333	0.08745
10	4	0.4	0.02078
...	...	...	...
13	4	0.30769	0.11309
14	4	0.28571	0.13507
15	5	0.33333	0.08745
16	5	0.3125	0.10828
17	5	0.29412	0.12666
18	5	0.27778	0.143

Таблица №2 для нач. случая i=2

n	$R(i,\ n)$	$\nu(i,\ n)$	$\Delta_n(i)$
1	0	0	0.57922
2	1	0.5	0.07922
3	2	0.66667	0.08745
4	2	0.5	0.07922

5	3	0.6	0.02078
6	4	0.66667	0.08745
7	5	0.71429	0.13507
8	5	0.625	0.04578
9	5	0.55556	0.02366
10	5	0.5	0.07922
...	...	...	...
62	35	0.56452	0.0147
63	35	0.55556	0.02366
64	36	0.5625	0.01672
65	37	0.56923	0.00999
66	38	0.57576	0.00346
67	39	0.58209	0.00287

Таблица №3 для нач. случая i=3

n	$R(i,\ n)$	$\nu(i,\ n)$	$\Delta_n(i)$
1	0	0	0

Литература по теории случайных процессов

1. Булинский А. В., А. Н. Ширяев А. Н. Теория случайных процессов: Учебник для вузов. — М.: ФИЗМАТЛИТ, 2005.
2. Вентцель Е. С., Овчаров Л. А. Теория случайных процессов и ее инженерные приложения: Учеб. пособие для вузов. — М.: Высшая школа, 2007.
3. Лобузов А.А., Гумляева С.Д., Норин Н.В. Задачи по теории случайных процессов. — М.: МИРЭА, 1993.
4. Письменный Д. Т. Конспект лекций по теории вероятностей, математической статистике и случайным процессам — М.: Айрис-пресс, 2007.
5. Прохоров А. В., Ушаков В. Г., Ушаков Н. Г. Задачи по теории вероятностей. Основные понятия, предельные теоремы, случайные процессы. — М.: КДУ, 2009.
6. Сборник задач по теории вероятностей, математической статистике и теории случайных функций: Учеб. пособие для вузов / Б.Г. Володин, М.П.Ганин, И.Я. Динер и др.; Под ред. А. А. Свешникова. — СПб.: Лань, 2008.
7. Кемени Д., Снелл Д. Конечные цепи Маркова. — М.: Наука, 1970.
8. Кемени Д., Снелл Д., Кнепп А. Счетные цепи Маркова. — М.: Наука, 1987.
9. Чжун Кай-Лай. Однородные цепи Маркова. — М.: Мир, 1964.
10. Карлин С. Основы теории случайных процессов. — М.: Мир, 1971.
11. Гихман И.И., Скороход А.В. Введение в теорию случайных процессов: Учеб. пособие для вузов. — М.: Наука, 1975.
12. Ивченко Г. И., Каштанов В. А., Коваленко И. Н. Теория массового обслуживания: Учеб. пособие для вузов. — М.: Либроком, 2012.

Приложение

```
import numpy as np

M = np.array([[0.437, 0.563, 0],
              [0.409, 0.591, 0],
              [0.235, 0.22, 0.545]])

arrayM = [M] # Array list P^n

def PrCheck (lastP, P, module = False):
  # PrCheck returns delta_n
  if module == False:
    return np.max(np.abs(np.array(P) - np.array(lastP)))
  else:
    # print("vec[i] =", P)
    return np.abs(P - lastP)




# 1 exercise
lastM = M
currentM = np.dot(M, M)
num = 0

print("Матрицы для 1 номера ------------------")
while PrCheck(lastM, currentM) >= 0.000001:
  arrayM.append(lastM)
  # print(np.round(lastM, 5), "\t", PrCheck(lastM, currentM))
  num += 1
  lastM = currentM
  currentM = np.dot(lastM, M)# P^n

print("---- 1 задание
-----------------------------------------------------------------")

# 2 exercise - сделать на бумажке

# решаем систему _p = _p * P1, где _p - стационарное распределение
# 0.437*p1 + 0.563*p2 + 0*p3 = p1
# 0.409*p1 + 0.591*p2 + 0*p3 = p2
# 0.235*p1 + 0.22*p2 + 0.545*p3 = p3
# p1 + p2 + p3 = 1

# преобразуем и тк ранг системы 2, одно можно выкинуть

# -0.563*p1 + 0.563*p2 + 0*p3 = 0
# 0.409*p1 - 0.409*p2 + 0*p3 = 0
# 0.235*p1 + 0.22*p2 - 0.455p*3 = 0
# p1 + p2 + p3 - 1 = 0

vec = np.array([409/972, 563/972, 0])

#print("vec =", vec)
#print("vec * M =", np.dot(vec, M))
#print()
```

```

print("3 exercise ----------------------------")
initialState = np.identity(3, dtype=float)
print(initialState)

arrayInitialState = []
lastInitialState = initialState

for i in range(3):
  print("\nNew vector")
  while PrCheck(lastInitialState[i], vec) >= 0.00001 :
    arrayInitialState.append(lastInitialState)
    print(lastInitialState[i], "\t", PrCheck(lastInitialState[i], vec), "\n")

    lastInitialState[i] = np.dot(lastInitialState[i], M)
  print(lastInitialState[i], "\t", PrCheck(lastInitialState[i], vec), "\n")
#print(lastInitialState)

# exercise 4

for i in range(1,4) :
  print("-------------------------------------------\ni = ", i)
  sequenceOfStates = [i]
  numberOfReturns = 0
  returnFrequency  = 0
  n = 1
  while PrCheck(returnFrequency, vec[i-1], module = True) >= 0.001 :
    memberOfTheSequence = np.random.random_sample()

    # вывод для таблицы
    print(n, "\t", round(numberOfReturns, 5), "\t", round(returnFrequency, 5), "\t",
round(PrCheck(returnFrequency, vec[i-1], module = True), 5))
    n+=1

    lastState = sequenceOfStates[len(sequenceOfStates) - 1]

    if memberOfTheSequence < M[lastState-1][0] :
      sequenceOfStates.append(1)

    elif memberOfTheSequence < M[lastState-1][0] + M[lastState-1][1] :
      sequenceOfStates.append(2)
    else :
      sequenceOfStates.append(3)
    numberOfReturns = sequenceOfStates.count(i) # возвращает количество чисел i,
встречающихся в последовательности
    returnFrequency = numberOfReturns / len(sequenceOfStates)



  print("Длина массива для нач. сост. (N_min(i))" +str(i)+ " = ",
len(sequenceOfStates)) # выводим длину массива N

```