|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | |  |  |  | | Министерство образования и науки РОССИЙСКОЙ ФЕДЕРАЦИИ | | |  Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»** | |
|  | |
|  | |
|  |  |

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 2

 по курсу «**Случайные процессы**»

Тема: **Марковский процесс с непрерывным временем и пятью состояниями**

Выполнил:

Студент 4-го курса

Петров С.В.

Группа: КМБО-03-17

МОСКВА 2020

**Задание**

1. Построить граф состояний марковского процесса
2. Написать систему дифференциальных уравнений Колмогорова для вероятностей состояний (с заданными интенсивностями)
3. Написать систему уравнений для нахождения стационарных вероятностей
4. Найти стационарное распределение вероятностей состояний (*, , , ,* )
5. Считая, что в начальным момент времени *t* = 0 система находится в состояний 1, провести моделирование развития системы до события с номером *K*, при котором впервые будет выполнено неравенство

где – относительная частота попадания системы в состояние i с 1-го по K событие (считаем, что при всех i), – число попаданий системы в состояние i в событиях с 1-го по K.

Событием считается переход системы из одного состояния в другое.

По результатам моделирования составить таблицу 1 с данными о событиях:

– номер события l;

– момент наступления события l;

– состояние C(l) системы после события l;

– время пребывания системы в состоянии C(l) с момента до перехода системы в другое состояние;

– значение отклонения

Если K > 100, то в отчет поместить только часть таблицы: данные о событиях с 1-го по 100, а также о событиях K-5, K-4, K-3, K-2, K-1, K.

1. Составить таблицу 2 с данными о состояниях следующего вида:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* | **1** | **2** | **3** | **4** | **5** |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

где

*i* – номер состояния;

– число попаданий системы в состояние *i* в событиях с 1-го по *K*;

– относительная частота попадания системы в состояние *i* в событиях с 1-го по *K*;

– общее время пребывания системы в состоянии *i* с момента до , т.е. до времени наступления события *K*+1

– доля времени пребывания системы в состоянии *i* в период с до

**Вычисления и вывод проводить с точностью до 0.00001.**

**Краткие теоретические сведения**

Случайный процесс , *t* ≥ 0 называется марковским, если для любого целого неотрицательного m, любых моментов времени *0 ≤ < < … < < s*, *t* > 0, любого набора состояний выполнено равенство

Процесс называется однородным (по времени), если условная вероятность перехода из состояния в состояние не зависит от *s*.

Эту вероятность обозначим через . В дальнейшем будем предполагать, что переходные вероятности дифференцируемы в нуле .

При *i ≠ j* число называется плотностью вероятности (интенсивностью) перехода из состояния в состояние .

Поведение процесса изучают с помощью ориентированного графа состояний, вершинами которого служат состояния системы, а стрелками обозначены возможные непосредственные переходы из состояния в состояние . Каждой стрелке приписана соответствующая плотность > 0 вероятности перехода.

Распределение вероятностей состояний, которое не зависит от времени для любых *i*, ≥ 0 и любых *i* = 1, 2, … называется стационарным.

Для вероятностей состояний справедлива система дифференциальных уравнений Колмогорова

Формулы для нахождения стационарного распределения:

Из системы линейных алгебраических уравнений, получающейся из системы дифференциальных уравнений Колмогорова, если положить в ней левые части равные нулю, а также добавить к ней нормировочное условие

Важной является задача о поведении вероятностей состояний системы при . Пределы, если они существуют, называются предельными вероятностями состояний.

Предельное распределение вероятностей, если оно существует и единственно, совпадает со стационарным.

Вычисления проводились средствами языка программирования Python:

Numpy – модуль для научных вычислений

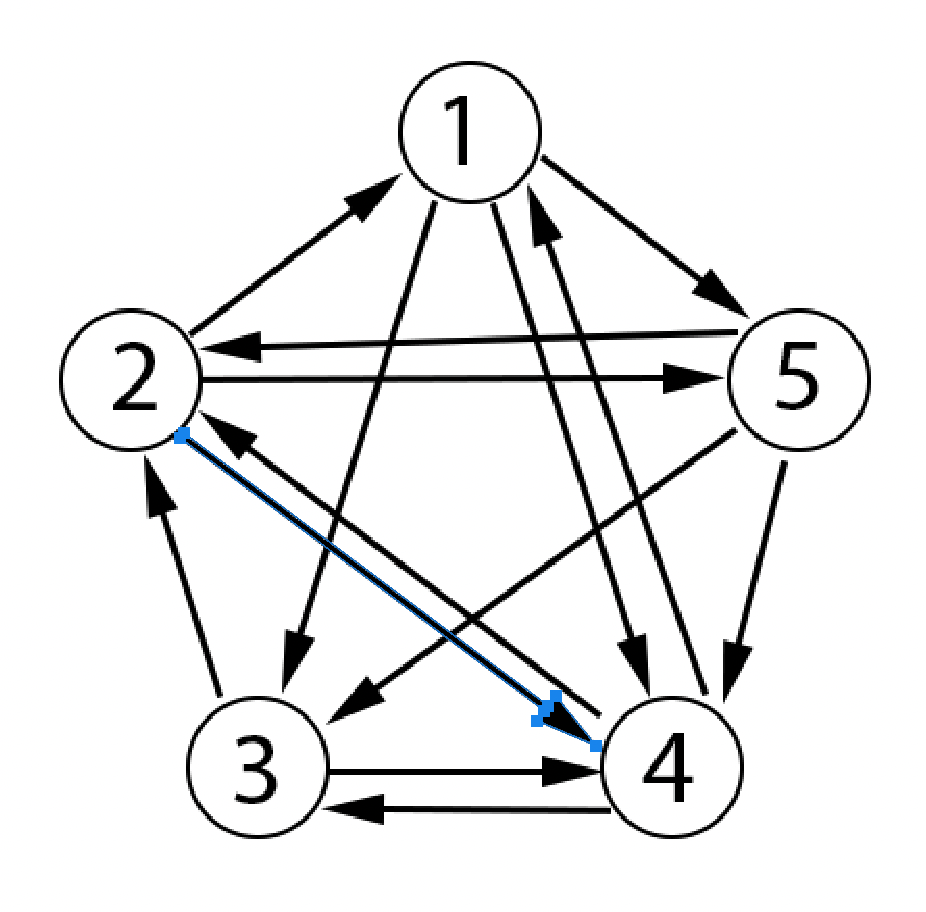
Copy – модуль для копирования объектов

Math – модуль с математическими функциями

**Результаты расчетов**

Вариант 40

1. Граф состояний:



1. Система дифференциальных уравнений Колмогорова для вероятностей состояний (с заданными интенсивностями):
2. Система уравнений для нахождения стационарных вероятностей:
3. Стационарное распределение вероятностей состояний:
4. Таблица 1 с данными о состояниях:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *№ события* |  | *C(l)* |  |  |
| 1 | 0.18839 | 5 | 1,41238 | 1 |
| 2 | 1.60077 | 4 | 2,62061 | 0.5 |
| 3 | 4.22138 | 2 | 2,31591 | 0.16667 |
| 4 | 6.5373 | 5 | 1,72121 | 0.08333 |
| 5 | 8.2585 | 2 | 3,87606 | 0.1 |
| 6 | 12.13457 | 1 | 2,32133 | 0.06667 |
| 7 | 14.45589 | 4 | 6,82318 | 0.04762 |
| 8 | 21.27908 | 3 | 1,13861 | 0.03571 |
| 9 | 22.41769 | 2 | 5,23962 | 0.02778 |
| 10 | 27.65731 | 1 | 12,75585 | 0.03333 |
| 11 | 40.41316 | 5 | 0,98767 | 0.02727 |
| 12 | 41.40084 | 3 | 7,49942 | 0.02273 |
| 13 | 48.90025 | 4 | 3,39533 | 0.01923 |
| 14 | 52.29558 | 2 | 8,74753 | 0.01648 |
| 15 | 61.04311 | 1 | 6,11436 | 0.01905 |
| 16 | 67.15747 | 5 | 0,15595 | 0.01667 |
| 17 | 67.31342 | 3 | 2,03128 | 0.01471 |
| 18 | 69.3447 | 4 | 1,23815 | 0.01307 |
| 19 | 70.58285 | 1 | 2,58297 | 0.0117 |
| 20 | 73.16582 | 4 | 1,55495 | 0.01053 |
| 21 | 74.72077 | 1 | 0,54062 | 0.0119 |
| 22 | 75.26139 | 4 | 9,34338 | 0.01082 |
| 23 | 84.60477 | 1 | 2,09033 | 0.01186 |
| 24 | 86.6951 | 3 | 0,78943 | 0.01087 |
| 25 | 87.48453 | 4 | 2,85999 | 0.01 |
| 26 | 90.34452 | 2 | 4,69186 | 0.01077 |
| 27 | 95.03638 | 1 | 3,0334 | 0.00997 |
| 28 | 98.06978 | 5 | 1,6783 | 0.00926 |
| 29 | 99.74808 | 3 | 3,70645 | 0.00862 |
| 30 | 103.45452 | 4 | 0,3513 | 0.00805 |
| 31 | 103.80582 | 1 | 0,08211 | 0.0086 |
| 32 | 103.88793 | 5 | 0,46142 | 0.00806 |
| 33 | 104.34935 | 3 | 2,49558 | 0.00758 |
| 34 | 106.84493 | 4 | 10,69446 | 0.00713 |
| 35 | 117.53938 | 2 | 2,24136 | 0.00756 |
| 36 | 119.78074 | 1 | 3,27561 | 0.00714 |
| 37 | 123.05636 | 4 | 0,05659 | 0.00676 |
| 38 | 123.11295 | 2 | 0,95613 | 0.00711 |
| 39 | 124.06908 | 1 | 4,34887 | 0.00675 |
| 40 | 128.41795 | 3 | 2,69655 | 0.00641 |
| 41 | 131.11449 | 2 | 2,28287 | 0.0061 |
| 42 | 133.39736 | 4 | 0,23618 | 0.00581 |
| 43 | 133.63354 | 1 | 2,15746 | 0.00609 |
| 44 | 135.791 | 5 | 6,19142 | 0.00581 |
| 45 | 141.98242 | 4 | 0,11716 | 0.00556 |
| 46 | 142.09958 | 2 | 3,00486 | 0.0058 |
| 47 | 145.10443 | 5 | 1,2604 | 0.00555 |
| 48 | 146.36483 | 4 | 3,45093 | 0.00532 |
| 49 | 149.81576 | 2 | 8,45823 | 0.00553 |
| 50 | 158.27398 | 5 | 0,13498 | 0.00531 |
| 51 | 158.40896 | 3 | 0,35472 | 0.0051 |
| 52 | 158.76368 | 4 | 3,225 | 0.0049 |
| 53 | 161.98868 | 1 | 1,06757 | 0.00508 |
| 54 | 163.05625 | 4 | 2,81291 | 0.00489 |
| 55 | 165.86916 | 3 | 1,23337 | 0.00505 |
| 56 | 167.10253 | 2 | 2,725 | 0.00487 |
| 57 | 169.82754 | 4 | 0,59443 | 0.0047 |
| 58 | 170.42197 | 2 | 0,79413 | 0.00484 |
| 59 | 171.21611 | 1 | 0,02425 | 0.00468 |
| 60 | 171.24036 | 3 | 1,44192 | 0.00452 |
| 61 | 172.68228 | 4 | 2,66148 | 0.00437 |
| 62 | 175.34376 | 3 | 0,71647 | 0.00449 |
| 63 | 176.06022 | 4 | 1,75142 | 0.00435 |
| 64 | 177.81165 | 3 | 4,74419 | 0.00446 |
| 65 | 182.55584 | 2 | 3,26822 | 0.00433 |
| 66 | 185.82406 | 1 | 2,90929 | 0.0042 |
| 67 | 188.73335 | 3 | 4,04928 | 0.00407 |
| 68 | 192.78263 | 4 | 0,01635 | 0.00395 |
| 69 | 192.79897 | 2 | 0,91291 | 0.00405 |
| 70 | 193.71188 | 1 | 2,59534 | 0.00393 |
| 71 | 196.30722 | 4 | 3,29731 | 0.00382 |
| 72 | 199.60453 | 1 | 0,23093 | 0.00391 |
| 73 | 199.83546 | 3 | 4,77598 | 0.00381 |
| 74 | 204.61144 | 4 | 7,11847 | 0.0037 |
| 75 | 211.72991 | 3 | 9,02062 | 0.00378 |
| 76 | 220.75053 | 2 | 4,30021 | 0.00368 |
| 77 | 225.05073 | 5 | 0,29517 | 0.00359 |
| 78 | 225.34591 | 3 | 3,93977 | 0.0035 |
| 79 | 229.28568 | 2 | 1,22628 | 0.00341 |
| 80 | 230.51196 | 5 | 5,34159 | 0.00332 |
| 81 | 235.85355 | 2 | 10,57016 | 0.00324 |
| 82 | 246.42371 | 5 | 0,12862 | 0.00316 |
| 83 | 246.55234 | 2 | 0,0141 | 0.00309 |
| 84 | 246.56644 | 1 | 0,34884 | 0.00301 |
| 85 | 246.91527 | 3 | 3,49096 | 0.00294 |
| 86 | 250.40624 | 2 | 2,67857 | 0.00287 |
| 87 | 253.0848 | 5 | 1,83745 | 0.00281 |
| 88 | 254.92225 | 2 | 0,08165 | 0.00274 |
| 89 | 255.00389 | 5 | 6,11915 | 0.00268 |
| 90 | 261.12304 | 2 | 7,42647 | 0.00262 |
| 91 | 268.54951 | 4 | 2,9984 | 0.00256 |
| 92 | 271.54791 | 1 | 2,11951 | 0.00263 |
| 93 | 273.66742 | 4 | 1,08407 | 0.00257 |
| 94 | 274.75149 | 2 | 7,62197 | 0.00263 |
| 95 | 282.37346 | 5 | 0,71227 | 0.00258 |
| 96 | 283.08573 | 3 | 7,15614 | 0.00252 |
| 97 | 290.24187 | 4 | 1,57103 | 0.00247 |
| 98 | 291.81289 | 2 | 3,27823 | 0.00252 |
| 99 | 295.09113 | 1 | 0,55015 | 0.00247 |
| 100 | 295.64128 | 5 | 2,04184 | 0.00242 |
| … | … | … | … | … |
| 283 | 802.09414 | 4 | 1,6362 | 0.00101 |
| 284 | 803.73034 | 3 | 0,9407 | 0.00102 |
| 285 | 804.67104 | 4 | 0,17697 | 0.00101 |
| 286 | 804.84801 | 3 | 3,18378 | 0.00102 |
| 287 | 808.03179 | 2 | 0,41601 | 0.00101 |
| 288 | 808.4478 | 1 | 7,10803 | 0.001 |

**Анализ результатов и выводы**

Таблица 2 с данными о состояниях:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* | **1** | **2** | **3** | **4** | **5** |  |
|  |  |  |  |  |  | 1 |
|  | 51 | 63 | 56 | 83 | 35 | 288 |
|  | 0.177708 | 0.21875 | 0.19444 | 0.28819 | 0.12153 | 1 |
|  | 161.47036 | 197.77465 | 146.39323 | 222.5971 | 87.13211 | 815.36744 |
|  | 0,198034 | 0,242559 | 0,179543 | 0,273002 | 0,106863 | 1 |

**Список литературы**

1. Лобузов А.А. Гумляева С.Д. Норин Н.В. Задачи по теории случайных процессов – М., 1993. – 68 с.
2. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и ее инженерные приложения. – М.: Высшая школа, 2007. – 428 с.
3. Лифшиц М.А. Случайные процессы – от теории к практике. – СПб.: Лань, 2016. – 320 с.

**Приложение**

**import** numpy **as** np

**import** copy

**from** math **import** fabs

A = [[-3, 0, 1, 1, 1], [1, -3, 0, 1, 1], [0, 1, -2, 1, 0], [1, 0, 1, -3, 1], [1, 1, 1, 0, -3]]

tmp = copy.deepcopy(A)

tmp = np.transpose(copy.deepcopy(A))

tmp = np.append(tmp, [[1, 1, 1, 1, 1]], axis=0)

**print**(tmp)

r = np.linalg.solve(tmp[1:6], [0, 0, 0, 0, 1])

**print**(r)

**print**("sum: ", sum(r))

**print**(np.matmul(r, A))

ri\_K = [0 **for** i **in** range(len(A))]

prev\_ri\_K = [0 **for** i **in** range(len(A))]

vi\_K = [0 **for** i **in** range(len(A))]

ti\_K = [0 **for** i **in** range(len(A))]

state\_time = np.random.exponential(scale= -1 / A[0][0])

**print**(state\_time)

t\_sob = [state\_time]

delta\_k = None

K = 1

state = 0

table1 = []

**while** (delta\_k **is** None) **or** (delta\_k > 0.001):

K += 1

k\_states = [i[0] **for** i **in** enumerate(A[state]) **if** i[1] > 0]

lambda\_i = -A[state][state]

alpha = np.random.random\_sample()

new\_state = 0

t = 1 / lambda\_i

**while** alpha > t:

new\_state += 1

t += (1 / lambda\_i)

new\_state = k\_states[new\_state]

ri\_K[new\_state] = ri\_K[new\_state] + 1

tau\_l = np.random.exponential(scale= - A[new\_state][new\_state])

t\_sob.append(t\_sob[-1] + tau\_l)

ti\_K[new\_state] += tau\_l

state = new\_state

delta\_k = max([fabs((ri\_K[i] / K) - (ri\_K[i] / (K - 1))) **for** i **in** range(len(A))])

delta\_i\_k = [ti\_K[i] / ((t\_sob[-1] + tau\_l) - t\_sob[0]) **for** i **in** range(len(A))]

**print**(delta\_i\_k)

vi\_K = [i / (K-1) **for** i **in** ri\_K]

**print**("rik: ", ri\_K)

**print**("vik: ", vi\_K)

**print**("tik: ", ti\_K)

**print**("t sob: ", t\_sob)

**print**(K)