



Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №2 по курсу "Моделирование"

Тема Марковские цепи

Студент Леонов В.В.

Группа ИУ7-76Б

Оценка (баллы) _____

Преподаватель Рудаков И.В.

Москва — 2022 г.

Формальная постановка задачи

Реализовать программное обеспечение для определения вероятности и времени пребывания системы в каждом из состояний в установившемся режиме работы СМО.

Исходные данные:

- количество состояний системы (до 10);
- матрица интенсивностей переходов из состояния в состояние.

Краткие теоритические сведения

Случайный процесс, протекающий в сложной системе S , называется **марковским**, если он обладает следующим свойством: для каждого момента времени t_0 вероятность любого состояния системы в будущем при $t > t_0$ зависит только от состояния системы в настоящем $t = t_0$ и не зависит от того, когда и каким образом система перешла в это состояние (как процесс развивался в прошлом).

В марковском случайном процессе будущее развитие зависит только от настоящего состояния и не зависит от предыстории процесса.

Для марковского процесса составлены уравнения Колмогорова:

$$F = (P'(t), P(t), \lambda) = 0$$

Вероятностью i -го состояния называется вероятность $P_i(t)$ того, что в момент времени t система будет находится в состоянии S_i . Для любого момента t сумма вероятностей всех состояний равно единице (правило нормировки).

Для нахождения предельных вероятностей используется система уравнений, определяемая следующим образом:

- в левой части каждого из уравнений стоит производная вероятности i -го состояния;

- в правой части - сумма произведений вероятностей всех состояний (из которых идут стрелки в данное состояние), умноженная на интенсивности соответствующих потоков событий, минус суммарная интенсивность всех потоков, выводящих систему из данного состояния, умноженная на вероятность данного i -го состояния.

После нахождения вероятностей, необходимо вычислить время пребывания системы в каждом из состояний. Для этого необходимо с заданным интервалом dt вычислять приращение вероятности для i -го состояния. Вычисления завершаются, когда найденная вероятность будет равна соответствующей предельной с точностью до заданной погрешности.

Для приращения вероятности dp необходимо задать начальные значения, например, $1/n$, где n — число состояний системы.

Средства реализации

Язык программирования — Python.

GUI — QT.

Листинг кода

```
1 from numpy import linalg
2 import random
3
4 TIME_DELTA = 1e-3
5 EPS = 1e-5
6
7 def random_mtr(size: int) -> list[list[float]]:
8     mtr = [[round(random.random(), 2) * random.randint(0, 1)
9              for _ in range(size)] for _ in range(size)]
10    return mtr
11
12
13 def get_coef_mtr(mtr: list[list[float]]) -> list[list[float]]:
14     size = len(mtr)
15     coef_mtr = [[0.0 for _ in range(size)] for _ in range(size)]
16
17     for i in range(size):
18         for j in range(size):
19             if (i == j):
20                 coef_mtr[i][i] = -sum(mtr[i]) + mtr[i][i]
21             else:
22                 coef_mtr[i][j] = mtr[j][i]
23     return coef_mtr
24
25
26 def calc_prob(mtr: list[list[float]]) -> list[float]:
27     size = len(mtr)
28     coef_mtr = get_coef_mtr(mtr)
29     coef_mtr[size - 1] = [1 for _ in range(size)]
30     ordinate_values = [0 if i != size - 1 else 1 for i in range(size)]
31     return linalg.solve(coef_mtr, ordinate_values).tolist()
32
33
34 def calc_prob_delta(mtr: list[list[float]], prob_curr: list[float]):
35     size = len(mtr)
36
37     prob_delta = []
38     coef_mtr = get_coef_mtr(mtr)
39
40     for i in range(size):
41         for j in range(size):
42             coef_mtr[i][j] *= prob_curr[j]
43         prob_delta.append(sum(coef_mtr[i]) * TIME_DELTA)
44     return prob_delta
```

```

1 def calc_time(mtr: list[list[float]], prob: list[float]):
2     size = len(mtr)
3
4     time_curr = 0.0
5     prob_curr = [1.0 / size for _ in range(size)]
6
7     time = [0.0 for _ in range(size)]
8     while not all(time):
9         prob_delta = calc_prob_delta(mtr, prob_curr)
10        for i in range(size):
11            if not time[i] and abs(prob_curr[i] - prob[i]) <= EPS:
12                time[i] = time_curr
13                prob_curr[i] += prob_delta[i]
14            time_curr += TIME_DELTA
15
16    return time
17
18
19 def calc_res(mtr: list[list[float]]) -> tuple[list[float], list[float]]:
20     prob = calc_prob(mtr)
21     time = calc_time(mtr, prob)
22     return prob, time

```

Демонстрация работы программы

Марковские цепи

Матрица интенсивностей

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 |
|-----|------|------|------|------|------|------|----|----|----|-----|
| S1 | 0.0 | 0.99 | 0.0 | 0.52 | 0.9 | 0.59 | | | | |
| S2 | 0.93 | 0.14 | 0.0 | 0.48 | 0.0 | 0.0 | | | | |
| S3 | 0.0 | 0.0 | 0.39 | 0.0 | 0.0 | 0.74 | | | | |
| S4 | 0.0 | 0.11 | 0.0 | 0.0 | 0.38 | 0.72 | | | | |
| S5 | 0.0 | 0.0 | 0.0 | 0.83 | 0.31 | 0.0 | | | | |
| S6 | 0.0 | 0.0 | 0.38 | 0.71 | 0.33 | 0.0 | | | | |
| S7 | | | | | | | | | | |
| S8 | | | | | | | | | | |
| S9 | | | | | | | | | | |
| S10 | | | | | | | | | | |

Количество состояний: 6

Случайное заполнение

Ответ

| | Вероятность P | Время t |
|-----|---------------|---------|
| S1 | 0.01038 | 9.11 |
| S2 | 0.03348 | 10.052 |
| S3 | 0.12238 | 2.748 |
| S4 | 0.33573 | 6.349 |
| S5 | 0.25971 | 7.027 |
| S6 | 0.23832 | 10.201 |
| S7 | | |
| S8 | | |
| S9 | | |
| S10 | | |

Вычислить

Очистить все

Рисунок 1 – Работа программы для системы с 6 состояниями

Марковские цепи

Матрица интенсивностей

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 |
|-----|------|------|------|------|------|------|------|------|------|------|
| S1 | 0.53 | 0.42 | 0.0 | 0.16 | 0.0 | 0.0 | 0.76 | 0.15 | 0.37 | 0.0 |
| S2 | 0.77 | 0.0 | 0.0 | 0.0 | 0.03 | 0.0 | 0.56 | 0.0 | 0.73 | 0.0 |
| S3 | 0.01 | 0.66 | 0.11 | 0.0 | 0.97 | 0.0 | 0.0 | 0.68 | 0.22 | 0.15 |
| S4 | 0.71 | 0.0 | 0.96 | 0.0 | 0.2 | 0.0 | 0.91 | 0.32 | 0.0 | 0.53 |
| S5 | 0.24 | 0.0 | 0.55 | 0.0 | 0.94 | 0.81 | 0.54 | 0.0 | 0.0 | 0.02 |
| S6 | 0.0 | 0.0 | 0.74 | 0.0 | 0.56 | 0.0 | 0.0 | 0.0 | 0.92 | 0.42 |
| S7 | 0.85 | 0.78 | 0.3 | 0.83 | 0.57 | 0.0 | 0.97 | 0.0 | 0.15 | 0.0 |
| S8 | 0.0 | 0.0 | 0.29 | 0.0 | 0.79 | 0.12 | 0.0 | 0.0 | 0.85 | 0.0 |
| S9 | 0.51 | 0.0 | 0.0 | 0.0 | 0.13 | 0.46 | 0.09 | 0.0 | 0.53 | 0.0 |
| S10 | 0.0 | 0.0 | 0.33 | 0.43 | 0.95 | 0.0 | 0.0 | 0.0 | 0.04 | 0.0 |

Количество состояний: 10

Случайное заполнение

Ответ

| | Вероятность P | Время t |
|-----|---------------|---------|
| S1 | 0.17274 | 7.342 |
| S2 | 0.09301 | 0.837 |
| S3 | 0.08088 | 6.885 |
| S4 | 0.03221 | 2.141 |
| S5 | 0.13263 | 7.377 |
| S6 | 0.08357 | 1.65 |
| S7 | 0.08776 | 0.374 |
| S8 | 0.0445 | 6.289 |
| S9 | 0.23445 | 7.355 |
| S10 | 0.03826 | 6.463 |

Вычислить

Очистить все

Рисунок 2 – Работа программы для системы с 10 состояниями

Марковские цепи

Матрица интенсивностей

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 |
|-----|------|------|------|----|----|----|----|----|----|-----|
| S1 | 0.82 | 0.65 | 0.0 | | | | | | | |
| S2 | 0.0 | 0.0 | 0.59 | | | | | | | |
| S3 | 0.02 | 0.0 | 0.0 | | | | | | | |
| S4 | | | | | | | | | | |
| S5 | | | | | | | | | | |
| S6 | | | | | | | | | | |
| S7 | | | | | | | | | | |
| S8 | | | | | | | | | | |
| S9 | | | | | | | | | | |
| S10 | | | | | | | | | | |

Количество состояний

3

Случайное заполнение

Ответ

| | Вероятность P | Время t |
|-----|---------------|---------|
| S1 | 0.0289 | 9.795 |
| S2 | 0.03184 | 19.039 |
| S3 | 0.93926 | 18.773 |
| S4 | | |
| S5 | | |
| S6 | | |
| S7 | | |
| S8 | | |
| S9 | | |
| S10 | | |

Вычислить

Очистить все

Рисунок 3 – Работа программы для системы с 3 состояниями